BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 March 2002 (14.03.2002)

PCT

(10) International Publication Number
WO 02/21702 A1

(51) International Patent Classification[7]: H03M 13/29, 13/27

(21) International Application Number: PCT/US01/27597

(22) International Filing Date:
5 September 2001 (05.09.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/230,578	5 September 2000 (05.09.2000)	US
60/232,053	12 September 2000 (12.09.2000)	US
60/232,288	12 September 2000 (12.09.2000)	US

(71) Applicant *(for all designated States except US)*: **BROADCOM CORPORATION** [US/US]; P.O. Box 57013, Irvine, CA 92619-7013 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CAMERON, Kelly, B.** [US/US]; P.O. Box 57013, Irvine, CA 92619-7013 (US). **JONES, Christopher, R.** [US/US]; P.O. Box 57013, Irvine, CA 92619-7013 (US).

(74) Agent: **PAULEY, Nicholas, J.**; Christie, Parker & Hale, P.O. Box 7068, Pasadena, CA 91105-7068 (US).

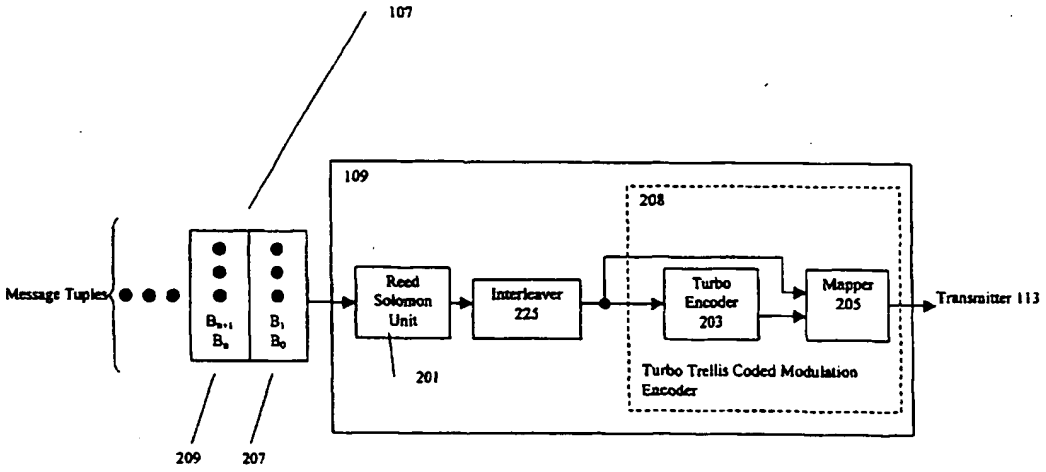
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: QUASI ERROR FREE (QEF) COMMUNICATION USING TURBO CODES



(57) Abstract: A method for encoding and decoding information using turbo codes. Turbo codes, though providing error performance superior to many codes, exhibit a phenomenon known as error floor. The error floor is a bit error level that turbo codes themselves have had difficulty improviding on. This error floor can prove problematical for quasi error free applications in which error rates on the order of $10._{10}$ are required. In addition turbo codes have failure modes in which errors occur in a burst, which may preclude the establishment of quasi error free communication. Such quasi error free applications also require an error correction that is superior to that generally represented by the error floor of turbo-codes. Quasi error free performance can be produced using turbo codes as an inner code in conjunction with an algebraic outer code linked by an interleaver. A combination of outer algebraic code and turbo inner code linked by an interleaver, which has a guaranteed minimum Depth between symbols output from the interleaver, can produce a quasi error free performance in systems utilizing Turbo-Codes without an increase in bandwidth.

WO 02/21702 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

QUASI ERROR FREE (QEF) COMMUNICATION USING TURBO CODES

FIELD OF THE INVENTION

The invention relates to methods, apparatus, and signals used in channel coding and decoding, and, in particular embodiments to methods, apparatus for use with a code having a turbo encoded inner code linked to an outer algebraic code via an interleaver.

BACKGROUND OF THE INVENTION

A significant amount of interest has recently been paid to channel coding. For example a recent authoritative text states:

"Channel coding refers to the class of signal transformations designed to improve communications performance by enabling the transmitted signals to better withstand the effects of various channel impairments, such as noise, interference, and fading. These signal-processing techniques can be thought of as vehicles for accomplishing desirable system trade-offs (e.g., error-performance versus bandwidth, power versus bandwidth). Why do you suppose channel coding has become such a popular way to bring about these beneficial effects? The use of large-scale integrated circuits (LSI) and high-speed digital signal processing (DSP) techniques have made it possible to provide as much as 10 dB performance improvement through these methods, at much less cost than through the use of most other methods such as higher power transmitters or larger antennas."

From "Digital Communications" Fundamentals and Applications Second Edition by Bernard Sklar, page 305 © 2001 Prentice Hall PTR.

Stated differently, improved coding techniques may provide systems that can operate at lower power, provide higher data rates, or provide lower bit error rates.

Turbo codes in particular have received recent attention due to their ability to obtain good performance over relatively noisy channels. For example bit error rates on the order of 10^{-5} to 10^{-6} can be achieved at signal to noise ratios less than a decibel away from modulation constrained capacities. However, their use in systems with Quasi Error Free (QEF) systems having residual BER requirements of 10^{-10} or lower, requires special consideration. This consideration is due partly to the presence of a characteristic of turbo codes known as the 'error floor' and partly due to a turbo code burst error mechanism.

Although Turbo-Codes may recover data at low signal to noise ratios, an increase in relative signal strength does not produce a corresponding drop in error rate. In fact

BERs of 10^{-6} to 10^{-7} may persist even for relatively large received signal to noise ratios. This problem is further discussed in a paper entitled "Investigating Quasi Error Free (QEF) Operation with Turbo Codes", K. Lakovic, C. Jones, J. Villasenor. International Symposium on Turbo Codes, Brest, France, September 4-7, 2000, and is incorporated by reference herein as though set forth in full. Accordingly the use of a turbo codes alone may not be sufficient to achieve QEF performance. There is a need in the art for systems that may be used with turbo codes and provide QEF type performance.

Conventions and Definitions:

Particular aspects of the invention disclosed herein depend upon and are sensitive to the sequence and ordering of data. To improve the clarity of this disclosure the following convention is adopted. Usually, items are listed in the order that they appear. Items listed as #1, #2, #3 are expected to appear in the order #1, #2, #3 listed, in agreement with the way they are read, i.e. from left to right. However, in engineering drawings, it is common to show a sequence being presented to a block of circuitry, with the right most tuple representing the earliest sequence, as shown in Figure 2B, where 207 is the earliest tuple, followed by tuple 209. The IEEE Standard Dictionary of Electrical and Electronics Terms, Sixth Edition, defines tuple as a suffix meaning an ordered set of terms (sequence) as in N-tuple. A tuple as used herein is merely a grouping of bits having a relationship to each other, such as for example a symbol.

Herein, the convention is adopted that items, such as tuples will be written in the same convention as the drawings. That is in the order that they sequentially proceed in a circuit. For example, "Tuples 207 and 209 are accepted by block 109" means tuple 207 is accepted first and then 209 is accepted, as is seen in Figure 2. In other words the text will reflect the sequence implied by the drawings. Therefore a description of Figure 2 would say "tuples 207 and 209 are provided to block 109" meaning that tuple 207 is provided to block 109 before tuple 209 is provided to block 109.

Herein an interleaver is defined as a device having an input and an output. The input accepting data tuples and the output providing data tuples having the same component bits as the input tuples, except for order.

An integral tuple (IT) interleaver is defined as an interleaver that reorders tuples that have been presented at the input, but does not separate the component bits of the input tuples. That is the tuples remain as integral units and adjacent bits in an input tuple will remain adjacent, even though the tuple has been relocated. The tuples, which are output from an IT interleaver are the same as the tuples input to interleaver, except for order. Hereinafter when the term interleaver is used, an IT interleaver will be meant.

A separable tuple (ST) interleaver is defined as an interleaver that reorders the

tuples input to it in the same manner as an IT interleaver, except that the bits in the input tuples are interleaved independently, so that bits that are adjacent to each other in an input tuple are interleaved separately and are interleaved into different output tuples. Each bit of an input tuple, when interleaved in an ST interleaver, will typically be found in a different tuple than the other bits of the input tuple from where it came. Although the input bits are interleaved separately in an ST interleaver, they are generally interleaved into the same position within the output tuple as they occupied within the input tuple. So for example, if an input tuple comprising two bits, a most significant bit and a least significant bit, is input into an ST interleaver the most significant bit will be interleaved into the most significant bit position in a first output tuple and the least significant bit will be interleaved into the least significant bit position in a second output tuple.

Modulo-N sequence designation is a term meaning the modulo-N of the position of an element in a sequence. If there are k item $s^{(i)}$ in a sequence then the items have ordinal numbers 0 to k-1, i.e. I_0 through $I_{(k-1)}$ representing the position of each time in the sequence. The first item in the sequence occupies position 0, the second item in a sequence I_1 occupies position 1, the third item in the sequence I_2 occupies position 2 and so forth up to item I_{k-1}, which occupies the k'th or last position in the sequence. The modulo-N sequence designation is equal to the position of the item in the sequence modulo-N . For example, the modulo-2 sequence designation of I_0=0, the modulo-2 sequence designation of I_1=1, and the modulo-2 sequence designation of I_2=0 and so forth.

A modulo-N interleaver is defined as an interleaver wherein the interleaving function depends on the modulo-N value of the tuple input to the interleaver. Modulo interleavers are further defined and illustrated herein.

A modulo-N encoding system is one that employs one or more modulo interleavers.

SUMMARY OF PREFERRED EMBODIMENTS OF THE INVENTION

In one aspect of the invention, the disclosure illustrates a method for providing quasi error free (QEF) encoding by algebraically encoding data, interleaving the encoded data in an interleaver having a guaranteed Depth, and turbo encoding the interleaved data.

In another aspect of the invention, the disclosure illustrates a method for decoding QEF encoded data by turbo decoding the QEF encoded data, deinterleaving the data using a deinterleaver having a guaranteed Depth and algebraically decoding

the deinterleaved data.

In a further aspect of the invention, the disclosure illustrates an apparatus for providing quasi error free (QEF) encoding. The apparatus includes an input that accepts data, an algebraic encoder that receives the data, an interleaver, which has a guaranteed depth, that interleaves the data and a turbo encoder that encodes the interleaved data.

In still a further aspect of the invention, the disclosure illustrates an apparatus for providing quasi error free (QEF) encoding. The apparatus includes a turbo decoder that accepts QEF data to be decoded, a deinterleaver that accepts the turbo decoded data and produce deinterleaved data and

an algebraic decoder that decodes the deinterleaved data.

BRIEF DESCRIPTION OF THE DRAWINGS

The features, aspects, and advantages of the present invention which have been described in the above summary will be better understood with regard to the following description, appended claims, and accompanying drawings where:

Figure 1 is a graphical illustration of an environment in which embodiments of the present invention may operate.

Figure 2A is a block diagram describing both the encoding and decoding in a turbo coded system according to an embodiment of the invention.

Figure 2B is a block diagram of a portion of a signal encoder according to an embodiment of the invention.

Figure 2C is a block diagram of a Ramsey type interleaver as may be used with embodiments of the present invention.

Figure 3 is a block diagram of a parallel concatenated (turbo) encoder, illustrating the difference between systematic and nonsystematic forms.

Figure 4 is a schematic diagram of a rate 2/3 "feed forward" convolutional nonsystematic encoder.

Figure 5 is a schematic diagram of a rate 2/3 "recursive" convolutional nonsystematic encoder.

Figure 6 is a trellis diagram of the convolutional encoder illustrated in Figure 5.

Figure 7 is a block diagram of a turbo-trellis coded modulation (TTCM) encoder.

Figure 8A is a block diagram of a TTCM encoder utilizing multiple interleavers.

Figure 8B is a graphical illustration of the process of modulo interleaving.

Figure 8C is a further graphical illustration of the process of modulo interleaving.

Figure 9 is a block diagram of a TTCM encoder employing a tuple interleaver.

Figure 10 is a block diagram of a TTCM encoder employing a bit interleaver.

Figure 11A is a first portion of combination block diagram and graphical illustration of a rate 2/3 TTCM encoder employing a ST interleaver, according to an embodiment of the invention.

Figure 11B is a second portion of combination block diagram and graphical illustration of a rate 2/3 TTCM encoder employing a ST interleaver, according to an embodiment of the invention.

Figure 12 is a combination block diagram and graphical illustration of rate ½ parallel concatenated encoder (PCE) employing a modulo-N Interleaver.

Figure 13 is a graphical illustration of the functioning of a modulo-4 ST interleaver, according to an embodiment of the invention.

Figure 14A is a graphical illustration of the generation of interleaver sequences from a seed interleaving sequence.

Figure 14B is a graphical illustration of a process by which modulo-2 and modulo-3 interleaving sequences may be generated.

Figure 14C is a graphical illustration of a process by which a modulo-4 interleaving sequence may be generated..

Figure 15 is a graphical illustration of trellis encoding.

Figure 16 is a graphical illustration of Turbo Trellis Coded Modulation (TTCM) encoding.

Figure 17 is a graphical illustration of a rate 2/3 TTCM encoder according to an embodiment of the invention.

Figure 18A is a graphical illustration of a rate ½ TTCM encoder, with constituent 2/3 rate encoders, according to an embodiment of the invention.

Figure 18B is a graphical illustration of alternate configurations of the rate ½ TTCM encoder illustrated in Figure 18A.

Figure 18C is a graphical illustration of alternate configurations of the rate ½ TTCM encoder illustrated in Figure 18A.

Figure 18D is a graphical illustration of alternate configurations of the rate ½ TTCM encoder illustrated in Figure 18A.

Figure 18E is a graphical illustration of alternate configurations of the rate ½ TTCM encoder illustrated in Figure 18A.

Figure 19 is a graphical illustration of a rate 3/4 TTCM encoder, with constituent 2/3 rate encoders, according to an embodiment of the invention.

Figure 20A is a graphical illustration of a rate 5/6 TTCM encoder, with constituent 2/3 rate encoders, according to an embodiment of the invention.

Figure 20B is a graphical illustration which represents an alternate encoding that will yield the same coding rate as Figure 20A.

Figure 21A is a graphical illustration of a rate 8/9 TTCM encoder, with constituent 2/3 rate encoders, according to an embodiment of the invention.

Figure 21B is a graphical illustration which represents an alternate encoding that will yield the same coding rate as Figure 21A

Figure 22 is a graphical illustration of map 0 according to an embodiment of the invention.

Figure 23 is a graphical illustration of map 1 according to an embodiment of the invention.

Figure 24 is a graphical illustration of map 2 according to an embodiment of the invention.

Figure 25 is a graphical illustration of map 3 according to an embodiment of the invention.

Figure 26 is a block diagram of a modulo-2 (even/odd) TTCM decoder according to an embodiment of the invention.

Figure 27 is a TTCM modulo-4 decoder according to an embodiment of the invention.

Figure 28 is a graphical illustration of a modulo-N encoder/decoder system according to an embodiment of the invention.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

Figure 1 is a graphic illustration of an environment in which embodiments of the present invention may operate. The environment illustrated at 101 is an information distribution system, such as may be found in a cable television distribution system.

In Figure 1 data is provided to the system from an information source 103. For purposes of illustration, the information source displayed in Figure 1 may be considered to be a cable television system head end which provides video data to end users. A formatter 105 accepts data from the information source 103. The data provided by information source 103 may comprise analog or digital signals such as (but not limited to) video signals, audio signals, and data signals. The formatter block 105 accepts the data from the information source and formats it into an appropriate form, such as message tuples, which are illustrated at 107. The formatted data is then provided to a channel encoder 109. Channel encoder 109 encodes that data provided to it. In some embodiments of the present invention, the channel encoder 109 may provide an encoding, with different goals depending on the particular implementation, for example

to make the signal more robust, to reduce the error probability, to operate the system using less transmission power or to enable a more efficient decoding of the signal. Channel encoder 109 then provides the encoded data to a transmitter 111. The transmitter transmits the encoded data provided to it by the channel encoder 109, for example, using an antenna 113. The signal broadcast from antenna 113 is received by a relay satellite 115 and then rebroadcast to a receiving terrestrial antenna, such as earth station antenna 117. Earth station antenna 117 collects the satellite signal and provides the collected signal to a receiver 119. The receiver 119 will amplify and demodulate/detect the signal as appropriate and provide the detected signal to a decoder 121. Decoder 121 will essentially, reverse the process of the channel encoder 109 and recreate the message tuples 123, which should represent a good estimate of the message tuples 107 that had been broadcast. The decoder 121 may use Forward Error Correction (FEC), in order to correct errors in the received signal. The tuples 123 provided by the decoder are then provided to a formatting unit 125, which prepares the received message tuples for use by an information sink, such as the television display illustrated at 127.

Figure 2A is a block diagram describing both the encoding and decoding in a turbo coded system according to an embodiment of the invention.

Data Source 221 provides data tuples to an algebraic encoder 223. Algebraic encoder 223 produces a stream of encoded codewords. The codewords produced by encoder 223 are provided to an interleaver 225. Interleaver 225 is an integral tuple (IT) interleaver, meaning that the bits of the symbols input are not rearranged at the output, only the order of the input symbols are rearranged. The interleaver 225 can be a variety of different types of interleavers, as will be discussed later. For the purposes of simplifying the disclosure interleaver 225 is illustrated hereinbelow as a Ramsey convolutional interleaver. The Ramsey convolutional interleaver was originally presented in a paper "Realization of Optimum Interleavers," by J.L. Ramsey, IEEE Transactions on Information Theory, Vol IT-16, May 1970, which is incorporated by reference herein as though set forth in full.

The interleaved sequence provided by interleaver 225 is turbo encoded in turbo encoder 227 and transmitted through either a wireline or wireless channel 229. Turbo decoding 231 follows at the received end. The decoded data provided by the turbo decoder 231 is provided to a de-interleaver 233 in a process that reverses the interleaving process 225. The de-interleaver assembles the received symbols back into the order in which they were originally produced by the algebraic encoder.

A particular interleaver implementation is next described in order to illustrate

diversification of the outer code such that long bursts of errors at the output of the inner (turbo) decoder do not preclude the overall coding system from achieving quasi error free operation.

Figure 2B is a block diagram of a portion of a signal encoder according to an embodiment of the invention. In Figure 2 message tuples 107 are provided to channel encoder 109. Channel encoder 109 comprises a Reed-Solomon unit 201, which provides a first encoding of the message tuples 107. The Reed-Solomon unit 201 comprises an exemplary encoder that may be used. Other types of encoders may also be used depending on the requirements of the implementations desired. The output of the Reed-Solomon (RS) unit 201 which includes a RS encoder and may include an interleaver is then provided to an interleaver 225. As discussed above the interleaver 225 is illustrated as a Ramsey interleaver. The type of interleaver used is not as important as the depth characteristics produced by the interleaver, and essentially any interleaver capable of providing a guaranteed minimum Depth may be used. Depth or distance between output symbols may be set at a desired level in order to guarantee that the burst errors of a certain length may be effectively dealt with. The depth necessary to guarantee that burst errors of a certain length may be repaired is a function of the depth of separation of input symbols. The greater the depth the larger the burst error that may be corrected. The mathematical relationship between depth and burst size correction is discussed below with respect to Figure 2C. The output of interleaver 225 is provided to a turbo trellis-coded modulation (TTCM) encoder 208. The interleaved output of the Reed-Solomon unit 201 is then provided to a turbo encoder 203, which applies a parallel concatenated (turbo) encoding to the input received from the interleaved output of the Reed-Solomon unit 201, and further provides it to a mapper 205. In addition, some of the bits of the interleaved data output from the Reed-Solomon unit 201 may bypass the turbo encoder 203 entirely and be coupled directly into the mapper 205.

The output of interleaver 225 is provides to a turbo trellis-coded modulation (TTCM) encoder 208. The output of the Reed-Solomon unit 201 is then provided to a turbo encoder 203, which applies a parallel concatenated (turbo) encoding to the input received from the Reed-Solomon unit 201, and further provides it to a mapper 205. In addition, some of the bits of the data output from the Reed-Solomon unit 201 may bypass the turbo encoder 203 entirely and be coupled directly into the mapper 205. Such data bits which bypasses the turbo encoder 203 are commonly referred to as uncoded bits. The uncoded bits are taken into account in the mapper 205 but are never actually encoded in the turbo encoder 203. In some embodiments of the invention

there are no uncoded bits. In other embodiments of the invention there may be several uncoded bits depending on the data rate of the overall turbo trellis-coded modulation (TTCM) encoder desired. The output of the Reed-Solomon unit 201 may vary in form depending on the overall rate desired from the TTCM encoder 208. Turbo encoders, such as that illustrated at 203, may have a variety of forms and classifications. One of the classifications of encoders in general and turbo encoders in particular is illustrated in Figure 3.

Figure 2C is a block diagram of a Ramsey type interleaver as may be used with embodiments of the present invention. T is equal to the number of burst error symbols to be corrected. An interleaving scheme which may guarantee a minimum Depth may achieve complete correction of an error burst of length up to and including T*Depth from the output of the inner decoder by the combination of de-interleaving and outer decoding. The illustrated system provides for correction of 20 or less errors per outer code block. The depth of the symbols provided by interleaver 225 is 201. The cost of such a scheme in terms of memory is given by Depth * BLen = 205824 outer code symbols, where Blen is the length, in terms of outer code symbols, of an outer code block and where outer code symbols are 10 bits wide. The outer coding scheme with the above listed parameters allows for complete correction of inner decoder burst errors of lengths up to 4020 outer code symbols.

The Ramsey interleaver 225 illustrated in figure 2C can be modeled as a ring of memory locations. Each memory location is sized so as to hold a bit tuple corresponding to the size of the symbols input to the interleaver 225.

Input symbols are placed in the ring Depth tuples apart. The ringsize is selected so that Depth * Blen (where Blen is equal to the length, in terms of the symbols input into the interleaver, of an outer code block.) is equal to the ring size. Also Depth and Blen are selected to be relatively prime numbers, which are not divisors of one another. The symbols are placed in the ring Depth apart until the ring is full. Once the ring is full the interleaver can output interleaved symbols by outputting sequential symbols from the ring as shown generally at 255.

In order to achieve quasi error free communications interleaver 225 guarantees that a particular Depth between output symbols will be maintained. Accordingly truly random interleavers will not work unless they are constrained so that a minimum Depth can be maintained.

The necessity of being able to correct burst errors in order to achieve a quasi error free communication using a system as illustrated in figure 2A is not at all obvious. Simulations have found that turbo codes exhibit a phenomenon in which certain data

patterns may cause burst errors of incorrect data. This phenomenon is rare and it does not significantly increase the overall error rate by a statistically meaningful level, however it may preclude the establishment of a quasi error free communication using Turbo Codes. Therefore in order to establish a QEF channel, and enable communications that require a QEF channel, an interleaver 225 is provided. Such a system may be used with any of the turbo encoding and modulo interleaving methods described hereinbelow.

Figure 3 is a block diagram of a parallel concatenated encoder illustrating the difference between systematic and nonsystematic forms. In Figure 3 data is input into the circuit at 301. Data is output from the parallel concatenated encoder (PCE) circuit 300 at 303. The data output 303 of the PCE illustrated at 300 may reach the output via three different paths. Input data tuples (groups of one or more bits) may be received at 301 and coupled directly to the data output 303 through selector mechanism 305 along the path labeled D. The data input may also be coupled into a first encoder 307 where it will be encoded and then coupled along the path E_1 through selector 305 and into data output 303. The data accepted into the PCE circuit at 301 may also be provided to an interleaver 309. Interleaver 309 rearranges the input sequence of the data accepted by the PCE circuit at 301. In other words, the interleaver shuffles the order of the data so that the data out of the interleaver 309 is not the same order as the data into the interleaver 309. The data out of the interleaver 309 is then provided to a second encoder 311. The second encoder 311 encodes the data provided to it by the interleaver 309 and then provides the encoded data along path E_2 through the selector 305 into the data output 303. If the selector 305 selects the data from path D and E_1 and E_2, where D represents all of the input data tuple, then a systematic-type turbo encoding is performed. However, if the data selector selects only between path E_1 and E_2, such that there is no direct path between the data input and data output, a nonsystematic turbo encoding is performed. In general the data input at 301 comprises input data tuples which are to be encoded. The data output at 303 comprises code words, which are the encoded representation of the input data tuples. In general, in a systematic type of encoding, the input tuples are used as part of the output code words to which they correspond. Within parallel concatenated encoders, such as that illustrated at 300, encoders such as the first encoder 307 and second encoder 311 are commonly referred to as component or constituent encoders because they provide encoding, which are used as components of the overall turbo encoding. The first encoder 307 and the second encoder 311 may also have a variety of forms and may be of a variety of types. For example, the first encoder 307 may be a block encoder or

a convolutional-type encoder. Additionally, the second encoder 311 may also be a block or convolutional-type encoder. The first and second encoders themselves may also be of systematic or nonsystematic form. The types of encoders may be mixed and matched so that, for example, the first encoder 307 may comprise a nonsystematic encoder and second encoder 311 may comprise a systematic-type encoder.

Constituent encoders, such as first encoder 307 and second encoder 311 may have delays incorporated within them. The delays within the encoders may be multiple clock period delays so that the data input to the encoder is operated on for several encoder clock cycles before the corresponding encoding appears at the output of the encoder.

One of the forms of a constituent encoder is illustrated in Figure 4.

Figure 4 is a schematic diagram of a rate two-thirds feed forward nonsystematic convolutional encoder. The encoder illustrated at 400 in Figure 4 is a rate two-thirds because there are two inputs 401 and 403 and three outputs 405, 407 and 409. Accordingly, for each input tuple comprising two input bits 401 and 403, which are accepted by the encoder 400, the output is a code word having three bits 405, 407 and 409. Therefore, for each two bits input at inputs 401 and 403 three bits are output at 405, 407 and 409. The encoder of Figure 4 comprises three delays 417, 419 and 421. Such delays may be formed from D-type flip flops or any other suitable delay or storage element. The rate two-thirds feed forward encoder of Figure 4 also comprises five modulo-2 adders 411, 413, 415, 423 and 425. Modulo-2 adders are adders in which the outputs of the modulo-2 adder is equal to the modulo-2 sum of the inputs. Delay elements 417, 419 and 421 are clocked by an encoder clock. Modulo-2 adders 411, 413, 415, 423 and 425 are combinational circuits which are unclocked. In combinational circuits the output appears a time delay after the inputs are changed. This time delay is due to the propagation time of the signal within the combinational circuits (this delay is assumed as a near zero delay herein) and not due to any clocking mechanisms. In contrast, a delay unit, such as 417, will not change its output until it receives an appropriate clock signal. Therefore, for an input signal to propagate, for example from input 403 through modulo-2 adder 411, through delay 417, through modulo-2 adder 413, through delay 419, through modulo-2 adder 415, through delay 421 in order to appear at output 409, the encoder clock 427 must first clock the input signal from 403 through delay unit 417, then through delay unit 419, and finally through delay unit 421. Therefore, once an input signal appears at 403 three encoder clocks 427 in succession will be required for the resultant output 409, which is associated with that input at 403, to be seen at the output.

The encoder of Figure 4 is a feed forward encoder. The signal is always fed forward and at no point in the circuit is there a path to feed back a signal from an later stage to an earlier stage. As a consequence a feed forward encoder, such as that illustrated in Figure 4, is a finite impulse response (FIR) type of state machine. That is, for an impulse signal applied at the input, the output will eventually settle into a stable state.

The encoder illustrated in Figure 4 may further be classified as a nonsystematic encoder because none of the inputs, that is either 401 or 403, appear at the output of the encoder. That is outputs 405, 407 or 409 don't reproduce the inputs in an encoded output associated with that input. This can be inferred from the fact that output 407, 405 and 409 have no direct connection to inputs 401 or 403.

Figure 5 is a schematic diagram of a rate two-thirds, recursive, convolutional nonsystematic encoder. The encoder of Figure 5 is similar to the encoder of Figure 4 in that both encoders are nonsystematic and convolutional. The encoder of Figure 5 is the same schematically as the encoder of Figure 4 with the addition of a third input at modulo-2 adder 511 and a third input at modulo-2 adder 515. The third input for each of modulo-2 adders 511 and 515 is formed by an additional modulo-2 adder 527. Modulo-2 adder 527 is formed in part by the output of delay 521. Modulo-2 adder 527 receives an input from delay 521 which is provided to modulo-2 adders 511 and 515. Accordingly the encoder of Figure 5 is recursive. In other words, the inputs of delays 517 and 521 are partially formed from outputs occurring later in the signal path and fed back to an earlier stage in the circuit. Recursive encoders may exhibit outputs that change when repeatedly clocked even when the inputs are held constant. The encoder of Figure 5 is a constituent encoder, and is used with an embodiment of the invention as will be described later. Figure 6 is a trellis diagram for the encoder illustrated in Figure 5. A trellis diagram is a shorthand method of defining the behavior of a finite state machine such as the basic constituent encoder illustrated in Figure 5. The state values in Figure 6 represent the state of the encoder. As can be seen from the trellis diagram in Figure 6, when the encoder of Figure 5 is in any single state, it may transition to any one of four different states. It may transition to four different states because there are two inputs to the encoder of Figure 5 resulting in four different possible input combinations which cause transitions. If there had been only one input to the encoder of Figure 5, for example, if inputs 501 and 503 were connected, then each state in the trellis diagram would have two possible transitions. As illustrated in the trellis diagram in Figure 6, if the encoder is in state 0, state 1, state 2 or state 3, the encoder may then transition into state 0, state 2, state 4 or state 6. However, if the

encoder is in state 4, state 5, state 6 or state 7, it may transition into state 1, state 3, state 5 or state 7.

Figure 7 is a block diagram of a turbo trellis-coded modulation (TTCM) encoder. In Figure 7 an input data sequence 701 is provided to an "odd" convolutional encoder 703 and an interleaver 705. The interleaver 705 interleaves the input data sequence 701 and then provides the resulting interleaved sequence to "even" convolutional encoder 707. Encoders 703 and 707 are termed "odd" and "even" respectively because encodings corresponding to odd input tuples (i.e. input tuple no. 1, 3, 5, etc.) are selected by selector 709 from encoder 703 and encodings corresponding to even input tuples (i.e. input tuple no. 0, 2, 4, etc.) are selected by selector 709 from encoder 707. The output of either the odd convolutional encoder 703 or the even convolutional encoder 707 is selected by a selecting mechanism 709 and then passed to a mapper 710. Figure 7 is a generalized diagram according to an embodiment of the invention which illustrates a general arrangement for a TTCM encoder. The odd convolutional encoder 703 receives the input data sequence and, in an embodiment of the invention, convolutionally, nonsystematically, encodes the input data sequence. Even convolutional encoder 707 receives the same input data as the odd convolutional encoder, except that the interleaver 705 has rearranged the order of the data. The odd and even convolutional encoders may be the same encoders, different encoders or even different types of encoders. For example, the odd convolutional encoder may be a nonsystematic encoder, whereas the even convolutional encoder may be a systematic encoder. In fact the convolutional encoders 703 and 707 may be replaced by block-type encoders such as Hamming encoders or other block-type encoders well known in the art. For the purposes of illustration, both constituent encoders 703 and 707 are depicted as nonsystematic, convolutional, recursive encoders as illustrated in Figure 5 . The select mechanism 709 selects, from convolutional encoder 703, outputs corresponding to odd tuples of the input data sequence 701. The select mechanism 709 selects, from convolutional encoder 707, outputs which correspond to even tuples of the input data sequence 701. Select mechanism 709 alternates in selecting symbols from the odd convolutional encoder 703 and the even convolutional encoder 707. The selector 709 provides the selected symbols to the mapper 710. The mapper 710 then maps the output of either the even convolutional encoder 707 or the odd convolutional coder 703 into a data constellation (not shown). In order to maintain a sequence made up of distance segments stemming from the even and odd input tuples, the selector 709 selects only encodings corresponding to even tuples of the input data sequence 701 from one encoder (e.g. 703), and selects only encoding corresponding to odd tuples of

the input data sequence from the other encoder (e.g. 707). This can be accomplished by synchronizing the selection of encoded tuples from the odd (703) and even (707) encoders, for example using a clock 711, and by using an odd/even interleaver 705 to maintain an even/odd ordering of input data tuples to the even encoder 707. The odd/even interleaver 705 will be described in detail later.

The encoder illustrated in Figure 7 is a type which will be known herein as a turbo trellis-coded modulation (TTCM) encoder. The interleaver 705, odd convolutional encoder 703, even convolutional encoder 707 and selector form a turbo encoder, also known as a parallel concatenated encoder (PCE). The encoder is known as a parallel concatenated encoder because two codings are carried on in parallel. For the parallel encoding, in the Figure 7 example one coding takes place in the odd convolutional encoder 703, and the other takes place in the even convolutional encoder 707. An output is selected sequentially from each encoder and the outputs are concatenated to form the output data stream. The mapper 710 shown in Figure 7 provides the trellis coded modulation (TCM) function. Hence, the addition of the mapper makes the encoder a turbo trellis-type encoder. As shown in Figure 7, the encoders may have any number of bits in the input data tuple. It is the topology that defines the encoder-type.

The encoder of Figure 7 is an illustration of only one of the possible configurations that may form embodiments of the present invention. For example, more than one interleaver may be employed, as shown in Figure 8.

Figure 8A is a block diagram of a TTCM encoder using multiple interleavers. Figure 8A illustrates an exemplary embodiment of the present invention utilizing N interleavers.

The first interleaver 802 is called the null interleaver or interleaver 1. Generally in embodiments of the invention the null interleaver will be as shown in Figure 8A, that is a straight through connection, i.e. a null interleaver. All interleaving in a system will be with respect to the null sequence produced by the null interleaver. In the case where the null interleaver is merely a straight through connection the null sequence out of the null interleaver will be the same as the input sequence. The concept of null interleaver is introduced as a matter of convenience, since embodiments of the invention may or may not have a first interleaver a convenient way to distinguish is to say "where the first interleaver is the null interleaver" when the first encoder receives input tuples directly and to say "where the first interleaver is an ST interleaver", when an ST interleaver occupies a position proximal to a first encoder.

In Figure 8A source input tuples 801 are provided to encoder 811 and to

interleavers 802 through 809. There are N interleavers counting the null interleaver as interleaver No. 1 and N encoders present in the illustration in Figure 8A. Other embodiments may additionally add an ST interleaver as interleaver No. 1 to process input tuples 801 prior to providing them to encoder 811.

Source tuples T_0, T_1 and T_2 are shown as three bit tuples for illustrative purposes. However, those skilled in the art will know that embodiments of the invention can be realized with a varying number of input bits in the tuples provided to the encoders. The number of input bits and rates of encoders 811 through 819 are implementation details and may be varied according to implementation needs without departing from scope and spirit of the invention.

Interleavers 803 through 809 in Figure 8A each receive the same source data symbols 801 and produce interleaved sequences 827 through 833. Interleaved sequences 827 through 833 are further coupled into encoders 813 through 819. Select mechanism 821 selects an encoded output from encoders 811 through 819. Selector 821 selects from each encoder 811 through 819 in sequence so that one encoded tuple is selected from each encoder in one of every N+1 selections. That is the selection number 0 (encoded tuple t_0) is chosen from encoder 811, the selection number 1 (encoded tuple u_1 is chosen from encoder 813 V_2 is chosen from encoder 815, and so forth. The same selection sequence is then repeated by selector 821.

In order not to miss any symbols, each interleaver is a modulo-type interleaver. To understand the meaning of the term modulo interleaver, one can consider the interleaver of Figure 7 as a modulo-2 interleaver. The interleaver of Figure 7 is considered a modulo-2 interleaver because input tuples provided to the interleaver during odd times (i.e. provided as input tuple 1, 3, 5 etc.) will be interleaved into odd time positions at the output of the interleaver (e.g. output tuple 77, 105, 321 etc.) That is the first tuple provided by an odd/even interleaver may be the third, fifth, seventh, etc. tuple provided from the interleaver, but not the second, fourth, sixth, etc. The result of any modulo-2 operation will either be a 0 or a 1, that is even or odd respectively, therefore the interleaver of Figure 7 is termed a modulo-2 or odd/even interleaver. In general, according to embodiments of the invention, the value of N for a modulo-N interleaving system is equal to the number of interleavers counting the Null interleaver as the first interleaver in the case where there is no actual first interleaver. The modulo interleaving system of Figure 8A is modulo-N because there are N interleaves, including null interleaver 1, interleaving system. The interleavers in a modulo interleaver system may interleave randomly, S randomly, using a block interleaver, or using any other mechanism for interleaving known in the art, with the additional restriction that

input/output positional integrity be maintained. When a sequence of tuples is interleaved, the modulo position value of an output will be the same as the modulo positional value of the input tuple. The position of a tuple modulo-N is known as a sequence designation, modulo designation, or modulo sequence designation. For example, in a modulo-4 interleaver the first tuple provided to the interleaver occupies position 0 of the input tuple stream. Because 0 modulo-4 is zero the modulo sequence designation of the first input tuple is 0 . The tuple occupying the position 0 may then be interleaved to a new output position #4, #8, #12, #16, etc., which also have the same modulo sequence designation, i.e. 0. The tuples occupying output position #4, #8, #12, #16 all have a sequence designation of 0 because 4 mod 4 = 8 mod 4 = 12 mod 4 = 16 mod 4 = 0. Similarly, the input tuple occupying position 2 and having sequence designation of 2 may be interleaved to a new output position #6, #10, #14, #20, etc, which also have the same modulo sequence designation of 2. The tuples in output positions #6, #10, #14, #20 etc have a modulo sequence designation of 2 because 6 mod 4 = 10 mod 4 = 14 mod 4 = 20 mod 4 = 2.

For example, in Figure 7 the modulo-2 interleaver 705, also known as an odd/even interleaver, may employ any type of interleaving scheme desired with the one caveat that the input data sequence is interleaved so that each odd sequence input to the interleaver is interleaved into another odd sequence at the output of the interleaver. Therefore, although interleaver 705 may be a random interleaver, it cannot interleave the inputs randomly to any output. It can, however, interleave any odd input to any random odd output and interleave any even input into any random even interleaved output. In embodiments of the present invention, a modulo interleaving system, such as that illustrated in Figure 8A, the interleavers must maintain the modulo positional integrity of interleaved tuples. For example, if there are 5 interleavers including the null interleaver (numbers 0-4) in Figure 8A, then Figure 8A would describe a modulo-5 interleaving system. In such a system, the input source data would be categorized by a modulo sequence number equal to the sequence position of the source data tuple modulo-5. Therefore, every input data tuple would have a sequence value assigned to it between 0 and 4 (modulo-5). In each of the 5 interleavers of the modulo-5 system, source data elements (characterized using modulo numbers) could be interleaved in any fashion, as long as they were interleaved into an output data tuple having an output sequence modulo number designation equal to the input sequence modulo number designation. The terms modulo sequence number sequence designation, modulo position value modulo designation, modulo position all refer to the same modulo ordering.

In other words an interleaver is a device that rearranges items in a sequence. The sequence is input in a certain order. An interleaver receives the items form the input sequence, I, in the order I_0, I_1, I_2, etc., I_0 being the first item received, I_1 being the second item received, item I_2 being the third item received. Performing a modulo-N operation on the subscript of I yields, the modulo-N position value of each input item. For example, if N=2 modulo-N position I_0 - $Mod_2(0)$ = 0 i.e. even, modulo-N position I_1 = $Mod_2(1)$ = 1 i.e., odd, modulo-N position I_2 = $Mod_2(2)$ = 0 i.e. even.

Figure 8B is a graphical illustration of examples of modulo interleaving. Interleaving is a process by which input data tuples are mapped to output data tuples.

Figure 8B illustrates of the process of modulo interleaving. As previously stated for the purposes of this disclosure an interleaver is defined as a device having one input and one output that receives a sequence of tuples and produces an output sequence having the same bit components as the input sequence except for order. That is, if the input sequence contains X bits having values of one, and Y bits having values of zero then the output sequence will also have X bits having values of 1 and Y bits having values of zero. An interleaver may reorder the input tuples or reorder the components of the input tuples or a combination of both. In embodiments of the invention the input and output tuples of an interleaver are assigned a modulo sequence designation which is the result of a modulo division of the input or output number of a tuple. That is, each input tuple is assigned a sequence identifier depending on the order in which it is accepted by the interleaver, and each output tuple is assigned a sequence identifier depending on the order in which it appears at the output of the interleaver.

For example, in the case of a modulo-2 interleaver the sequence designation may be even and odd tuples as illustrated at 850 in Figure 8B. In such an example, the input tuple in the 0 position, indicating that it was the first tuple provided, is designated as an even tuple T_0. Tuple T_1, which is provided after tuple T_0 is designated as an odd tuple, tuple T_2, which is provided after T_1 is designated as an even tuple and so forth. The result of the modulo interleaving is illustrated at 852. The input tuples received in order T_0, T_1, T_2, T_3, T_5, T_6 have been reordered to T_2, T_3, T_6, T_5, T_0, T_1, T_4. Along with the reordered tuples at 852 is the new designation I_0 through I_6 which illustrates the modulo sequence position of the interleaved tuples.

The modulo-2 type interleaver illustrated in figure 8B at 854 can be any type of interleaver, for example, a block interleaver, a shuffle interleaver or any other type of interleaver known in the art if it satisfies the additional constraint that input tuples are interleaved to positions in the output sequence that have the modulo position value.

Therefore an input tuple having an even modulo sequence designation will always be interleaved to an output tuple having an even modulo sequence designation and never will be interleaved to an output tuple having an odd modulo sequence designation. A modulo-3 interleaver 856 will function similarly to a modulo-2 interleaver 854 except that the modulo sequence designation will not be even and odd but zero, one and two. The sequence designation is formed by taking the modulo-3 value of the input position (beginning with input position 0. Referring to Figure 8B modulo-3 interleaver 856 accepts input sequence T_0, T_1, T_2, T_3, T_4, T_5 and T_6 (858) and interleaves it to interleaved sequence 860: T_3, T_4, T_5, T_6, T_1, T_2 which are also designated as interleaved tuples I_0 through I_6.

As a further illustration of modulo interleaving, a modulo-8 interleaver is illustrated at 862. The modulo 8 interleaver at 862 takes an input sequence illustrated at 864 and produces an output sequence illustrated at 866. The input sequence is given the modulo sequence designations of 0 through 7 which is the input tuple number modulo-8. Similarly, the interleaved sequence is given a modulo sequence designation equal to the interleaved tuple number modulo-8 and reordered compared to the input sequence under the constraint that the new position of each output tuple has the same modulo-8 sequence designation value as its corresponding input tuple.

In summary, a modulo interleaver accepts a sequence of input tuples which has a modulo sequence designation equal to the input tuple number modulo-N where N = H of the interleaver counting the null interleaver. The modulo interleaver then produces an interleaved sequence which also has a sequence designation equal to the interleaved tuple number divided by the modulo of the interleaver. In a modulo interleaver bits which start out in an input tuple with a certain sequence designation must end up in an interleaved modulo designation in embodiments of the present invention. Each of the N interleavers in a modulo N interleaving system would provide for the permuting of tuples in a manner similar to the examples in Figure 86; however, each (interleaver would yield a different permutation.

The input tuple of an interleaver, can have any number of bits including a single bit. In the case where a single bit is designated as the input tuple, the modulo interleaver may be called a bit interleaver.

Inputs to interleavers may also be arbitrarily divided into tuples. For example, if 4 bits are input to in interleaver at a time then the 4 bits may be regarded as a single input tuple, two 2 bit input tuples or four 1 bit input tuples. For the purposes of clarity of the present application if 4 bits are input into an interleaver the 4 bits are generally considered to be a single input tuple of 4 bits. The 4 bits however may also be

considered to be ½ of an 8 bit input tuple, two 2 bit input tuples or four 1 bit input tuples the principles described herein. If all input bits input to the interleaver are kept together and interleaved then the modulo interleaver is designated a tuple interleaver (a.k.a. integral tuple interleaver) because the input bits are interleaved as a single tuple. The input bits may be also interleaved as separate tuples. Additionally, a hybrid scheme may be implimented in which the input tuples are interleaved as tuples to their appropriate sequence positions, but additionally the bits of the input tuples are interleaved separately. This hybrid scheme has been designated as an ST interleaver. In an ST interleaver, input tuples with a given modulo sequence designation are still interleaved to interleaved tuples of similar sequence designations. Additionally, however, the individual bits of the input tuple may be separated and interleaved into different interleaved tuples (the interleaved tuples must all have the same modulo sequence designation as the input tuple from which the interleaved tuple bits were obtained). The concepts of a tuple modulo interleaver, a bit modulo interleaver, and a bit-tuple modulo interleaver are illustrated in the following drawings.

Figure 9 is a block diagram of TTCM encoder employing a tuple type interleaver. In Figure 9 an exemplary input data sequence 901 comprises a sequence of data tuples T_0, T_1, T_2, T_3 and T_4. The tuples are provided in an order such that T_0 is provided first, T_1 is provided second, etc. Interleaver 915 interleaves data sequence 901. The output of the interleaver comprises a new data sequence of the same input tuples but in different order. The data sequence 903, after interleaving, comprises the data tuples T_4, T_3, T_0, T_1 and T_2 in that order. The tuple interleaver illustrated in Figure 9 at 915 is a modulo-2 or odd/even type interleaver. The original data sequence 901 is provided to odd convolutional encoder 905 and the interleaved data sequence 903 is provided to an even convolutional encoder 907. A select mechanism 909 selects encoded outputs from the odd convolutional encoder 905 and the even convolutional encoder 907, according to the procedure provided below and illustrated in Figure 9, and provides the encoder output selected to the mapper 911. The select mechanism 909 illustratively chooses encoded outputs from the "odd" convolutional encoder 905 that correspond to odd tuples in the input data sequence 901. The select device 909 also chooses encoded tuples from the even convolutional encoder 907, that correspond to the even tuples of input sequence 903. So if the odd convolutional encoder 905 produces encoded tuples O_0, O_1, O_2, O_3 and O_4 corresponding to the input sequence of data tuples 901, the selector will select O_1 and O_3 (which have an odd modulo sequence designation) to pass through the mapper. In like manner if the even convolutional encoder produces symbols E_4, E_3, E_0, E_1 and E_2 from the input sequence

903 and select mechanism 909 selects E_4, E_0 and E_2 and passes those encoded tuples to the mapper 911. The mapper will then receive a composite data stream corresponding to encoded outputs E_4, O_1, E_0, O_3, and E_2. In this manner an encoded version of each of the input data sequence tuples 901 is passed onto the mapper 911. Accordingly, all of the input data sequence tuples 901 are represented in encoded form in the data 913 which is passed onto the mapper 911. Although both encoders encode every input tuple, the encoded tuples having an odd sequence designation are selected from encoder 905 and the encoded tuples having an even sequence designation are selected from encoder 907. In the interleaver 915 of Figure 9, each tuple is maintained as an integral tuple and there is no dividing of the bits which form the tuple. A contrasting situation is illustrated in Figure 10.

Figure 10 is a block diagram of a TTCM encoder employing a bit type interleaver. In Figure 10 an input tuple is represented at 1003 as input bits i_0 through i_{k-1}. The input bits i_0 through i_{k-1} are coupled into an upper constituent encoder of 1007. The input tuple 1003 is also coupled into interleaver 1005. The interleaver 1005 is further divided into interleavers 1009, 1011 and 1013. Each of the interleavers 1009, 1011 and 1013 accepts a single bit of the input tuple. The input tuple 1003 is then rearranged in the interleaver 1005 such that each bit occupies a new position in the sequence that is coupled into the lower constituent encoder 1015. The interleaving performed by the interleaver 1005 may be any type of suitable interleaving. For example, the interleaver may be a block interleaver a modulo interleaver as previously described, or any other type of interleaver as known in the art.

In the illustrated interleaver of Figure 10 the interleaving sequence provided by interleaver 1005, and hence by sub-interleavers 1009, 1011 and 1013, is independent of the positions of the bits within the input 1003. Input tuple 1001 represents input bits which are not passed through either of the constituent encoders 1007 or 1015. The upper encoding 1017 comprises the uncoded input tuple 1001 plus the encoded version of input tuple 1003, which has been encoded in the upper constituent encoder 1007. The lower encoding 1019 comprises the uncoded input tuple 1001 plus the output of the lower constituent encoder 1015 which accepts the interleaved version of input tuple 1003. A selector 1021 accepts either the upper or lower encoding and passes selected encoding to a symbol mapper 1023.

Figure 11A is a first part of a combination block diagram and graphic illustration of a rate 2/3 TTCM encoder employing a ST interleaver according to an embodiment of the invention. Figure 11A and 11B in combination illustrate a modulo-2 ST interleaver as may be used with a rate 2/3 TTCM encoder. In Figure 11A input tuples

1101 are provided to a rate 2/3 encoder 1103. The rate 2/3 encoder 1103 is designated as an even encoder because, although it will encode every input tuple, only the tuples corresponding to encoded even tuples will be selected from encoder 1103 by the selection circuit. Input tuples comprise 2 bits, a most significant bit designated by an M designation and a least significant bit designated by an L designation. The first tuple that will be accepted by the rate 2/3 even encoder 1103 will be the even tuple 1105. The even input tuple 1105 comprises 2 bits where M_0 is the most significant bit, and L_0 is the least significant bit. The second tuple to be accepted by the rate 2/3 even encoder 1103 is the 1107 tuple. The 1107 tuple is designated as an odd tuple and comprises a most significant bit M_1 and a least significant bit L_1. The input tuples are designated even and odd because the interleaver 1109, which is being illustrated in Figure 11A, is modulo-2 interleaver also known as an even/odd interleaver. The same principles, however, apply to any modulo-N interleaver. If the modulo interleaver had been a mod 3 interleaver instead of a mod 2 interleaver then the input tuples would have sequence designations 0, 1 and 2. If the modulo interleaver had been a modulo-4 interleaver then the input tuples would have modulo sequence designations 0, 1, 2, 3. The modulo interleaving scheme, discussed here with respect to modulo-2 interleavers and 2 bit tuples, may be used with any size of input tuple as well as any modulo-N interleaver. Additionally, any rate encoder 1103 and any type encoder may be used with the modulo ST interleaving scheme to be described. A rate 2/3 encoder, a modulo-2 ST interleaver, and 2 bit input tuples have been chosen for ease of illustration but are not intended to limit embodiments of the invention to the form disclosed. In other words, the following modulo-2 ST interleaver is chosen along with 2 bit input tuples and a rate 2/3 encoder system in order to provide for a relatively uncluttered illustration of the principles involved. The ST interleaver 1109 in this case actually can be conceptualized as two separate bit type interleavers 1111 and 1113. The separation of the interleavers is done for conceptual type purposes in order to make the illustration of the concepts disclosed easier to follow. In an actual implementation the interleaver 1109 may be implimented in a single circuit or multiple circuits depending on the needs of that particular implementation. Interleaver 1111 accepts the least significant bits of the input tuple pairs 1101. Note input tuple pairs designate input tuples having a pair, i.e. MSB and LSB, of bits. The interleaver 1111 interleaves the least significant bits of the input tuple pairs 1101 and provides an interleaved sequence of least significant bits of the input tuple pairs for example those illustrated in 1115. In the example, only eight input tuple pairs are depicted for illustration purposes. In an actual implementation the number of tuple pairs in a block to be interleaved could number tens of thousands or

even more. Eight input tuple pairs are used for ease of illustration purposes. The least significant bits of the input tuple pairs 1101 are accepted by the interleaver 1111 in the order L_0, L_1, L_2, L_3, L_4, L_5, L_6, and L_7. The interleaver, in the example of Figure 11A, then provides an interleaved sequence 1115 in which the least significant bits of the input tuples have been arranged in the order L_6, L_5, L_4, L_1, L_2, L_7, L_0 and L_3. Note that although the least significant bit of the input tuple pairs have been shuffled by the interleaver 1111 each least significant bit in an even tuple in the input tuple pairs is interleaved to an even interleaved position in the output sequence 1115. In like manner, odd least significant bits in the input sequence 1101 are interleaved by interleaver 1111 into odd position in the output sequence 1115. This is also a characteristic of modulo ST interleaving. That is although the data input is interleaved, and the interleaving may be done by a variety of different interleaving schemes know in the art, the interleaving scheme, however, is modified such that even data elements are interleaved to even data elements and odd data elements are interleaved to odd data elements. In general, in modulo-N interleavers the data input to an interleaver would be interleaved to output positions having the same modulo sequence designation as the corresponding modulo sequence designation in the input sequence. That is, in a modulo-4 interleaver an input data element residing in a input tuple with a modulo sequence designation of 3 would end up residing in an interleaved output sequence with a modulo sequence designation of 3. In other words, no matter what type of interleaving scheme the interleaver (such as 1111) uses, the modulo sequence designation of each bit of the input data tuples sequence is maintained in the output sequence. That is, although the positions of the input sequence tuples are changed the modulo interleaved positions are maintained throughout the process. This modulo sequence designation, here even and odd because a modulo-2 interleaver is being illustrated, will be used by the selection mechanism to select encoded tuples corresponding to the modulo sequence designation of the input tuples. In other words, the modulo sequence designation is maintained both through the interleavers and through the encoders. Of course, since the input tuples are encoded the encoded representation of the tuples appearing at the output of the encoder may be completely different and may have more bits than the input tuples accepted by the encoder.

Similarly, the most significant bits of input tuples 1101 are interleaved in interleaver 1113. In the example of Figure 11A, the sequence M_0 through M_7 is interleaved into an output sequence M_2, M_7, M_0, M_5, M_6, M_3, M_4, and M_1. The interleaved sequence 1117, produced by interleaving the most significant bits of the input tuples 1101 in interleaver 1113, along with the interleaved sequence of least significant bits

1115 is provided to into the "odd" rate 2/3 encoder 1119. Note that in both cases all data bits are interleaved into new positions which have the same modulo sequence designation as the corresponding input tuples modulo sequence designation.

Figure 11B is a second part of a combination block diagram and graphic illustration of a rate 2/3 TTCM encoder employing an ST interleaver. In Figure 11B the even rate 2/3 encoder 1103 and the odd rate 2/3 encoder 1119, as well as the tuples input to the encoders, are reproduced for clarity. Even encoder 1103 accepts the input tuple sequence 1101. The odd encoder 1119 accepts an input sequence of tuples, which is formed from the interleaved sequence of most significant bits 1117 combined with the interleaved sequence of least significant bits 1115. Both encoders 1103 and 1119 are illustrated as rate 2/3 nonsystematic convolutional encoders and therefore each have a 3 bit output. Encoder 1119 produces an output sequence 1153. Encoder 1103 produces an output sequence 1151. Both sequences 1151 and 1153 are shown in script form in order to indicate that they are encoded sequences. Both rate 2/3 encoders accept 2 bit input tuples and produce 3 bit output tuples. The encoded sequences of Figure 11B may appear to have 2 bit elements, but in fact the two letter designation and comprise 3 encoded bits each. Therefore, output tuple 1155 which is part of sequence 1153 is a 3 bit tuple. The 3 bit tuple 1155 however, is designated by a script M_7 and a script L_5 indicating that that output tuple corresponds to an input tuple 1160, which is formed from most significant bit M_7 and least significant bit L_5. In like manner, output tuple 1157 of sequence 1151 comprises 3 bits. The designation of output tuple 1157 as M_0 and L_0 indicates that that output tuple corresponds to the input tuple 1101, which is composed of input most significant bit M_0 and input least significant bit L_0. It is worthwhile to note that output tuple of encoder 1103, which corresponds to input tuple 1161 maintains the same even designation as input tuple 1161. In other words, the output tuple of an encoder in a modulo interleaving system maintains the same modulo sequence designation as the input tuple to which it corresponds. Additionally, in a ST interleaver input tuple bits are interleaved independently but are always interleaved to tuples having the same modulo sequence designation.

Selector mechanism 1163 selects between sequences 1153 and 1151. Selector 1163 selects tuples corresponding to an even modulo sequence designation from the sequence 1151 and selects tuples corresponding to an odd modulo sequence designation from sequence 1153. The output sequence created by such a selection process is shown at 1165. This output sequence is then coupled into mapper 1167. The modulo sequence 1165 corresponds to encoded tuples with an even modulo sequence designation selected from sequence 1151 and encoded tuples with an odd

modulo sequence designation selected from 1153. The even tuples selected are tuple M_0 L_0, tuple M_2 L_2, tuple M_4 L_4 and tuple M_6 L_6. Output sequence also comprises output tuples corresponding to odd modulo sequence designation M_7 L_5, tuple M_5 L_1, tuple M_3 L_7 and tuple M_1 and L_3.

A feature of modulo tuple interleaving systems, as well as a modulo ST interleaving systems is that encoded versions of all the input tuple bits appear in an output tuple stream. This is illustrated in output sequence 1165, which contains encoded versions of every bit of every tuple provided in the input tuple sequence 1101.

Those skilled in the art will realize that the scheme disclosed with respect to Figures 11A and 11B can be easily extended to a number of interleavers as shown in Figure 8A. In such a case, multiple modulo interleavers may be used. Such interleavers may be modulo tuple interleavers in which the tuples that will be coupled to the encoders are interleaved as tuples or the interleavers may be ST interleavers wherein the input tuples are interleaved to the same modulo sequence designation in the output tuples but the bits are interleaved separately so that the output tuples of the interleavers will correspond to different bits than the input sequence. By interleaving tuples and bits within tuples a more effective interleaving may be obtained because both bits and tuples are interleaved. Additionally, the system illustrated in Figures 11A and 11B comprise an encoder 1103 which accepts the sequence of input tuples 1101. The configuration of Figure 11A and 11B illustrates one embodiment. In a second embodiment the input tuples are ST interleaved before being provided to either encoder. In this way both the even and odd encoders can receive tuples which have had their component bits interleaved, thus forming an interleaving which may be more effective. In such a manner, an even encoder may produce a code which also benefits from IT or ST tuple interleaving. Therefore, in a second illustrative embodiment of the invention the input tuples are modulo interleaved before being passed to either encoder. The modulo interleaving may be a tuple interleaving, or a ST interleaving. Additionally, the types of interleaving can be mixed and matched.

Additionally, the selection of even and odd encoders is arbitrary and although the even encoder is shown as receiving uninterleaved tuples, it would be equivalent to switch encoders and have the odd encoder receive uninterleaved tuples. Additionally, as previously mentioned the tuples provided to both encoders may be interleaved.

Figure 12 is a combination block diagram and graphical illustration of a rate ½ parallel concatenated encoder (PCE) employing a modulo-N interleaver. Figure 12 is provided for further illustration of the concept of modulo interleaving. Figure 12 is an illustration of a parallel concatenated encoder with rate ½ constituent encoders 1207

and 1209. The input tuples to the encoder 1201 are provided to rate ½ encoder 1207. Each input tuple, for example, T_0, T_1, T_2 and T_n given an input tuple number corresponding to the order in which it is provided to the encoder 1207 and interleaver 1211. The input tuple number corresponds to the subscript of the input tuple. For example, T_0 the zero tuple is the first tuple provided to the rate ½ encoder 1207, T_1 is the second tuple provided to the rate ½ encoder 1207, T_2 is the third tuple provided to the rate ½ input encoder 1207 and T_n is the N plus first tuple provided to the rate ½ encoder 1207. The input tuples may be a single bit in which case the output of the rate ½ encoder 1207 would comprise 2 bits. The input tuples may also comprise any number of input bits depending on the number of inputs to the rate ½ encoder 1207. The modulo concept illustrated is identical where the rate ½ encoder is provided with tuples having a single bit or multiple bits. The input tuples 1201 are assigned a modulo sequence designation 1205. The modulo sequence designation is formed by taking the input tuple number modulo-N, which is the modulo order of the interleaver. In the example illustrated, the modulo order of the interleaver 1211 is N. Because the modulo order of the interleaver is N the modulo sequence designation can be any integer value between 0 and N-1. Therefore, the T_0 tuple has a modulo sequence designation of 0, the T_1 tuple has a modulo sequence designation of 1, the T_{n-1} input tuple has a modulo sequence designation of N-1, the T_n input tuple has a modulo sequence designation of 0 and the T_{n+1} input tuple has a modulo sequence designation of 1 and so forth. Interleaver 1211 produces interleaved tuples 1215. Similarly to the input tuples the interleaved tuples are given a modulo sequence designation which is the same modulo order as the interleaver 1211. Therefore, if the input tuples have a modulo sequence designation from 0 to N-1 then the interleaved tuples will have a modulo sequence designation of 0 to N-1. The interleaver 1211 can interleave according to a number of interleaving schemes known in the art. In order to be a modulo interleaver, however, each of the interleaving schemes must be modified so that input tuples with a particular modulo sequence designation are interleaved to interleaved tuples with the same modulo sequence designation. The interleaved tuples are then provided to a second rate ½ encoder 1209. The encoder 1207 encodes the input tuples, the encoder 1209 encodes the interleaved tuples and selector 1219 selects between the output of the encoder 1207 and the output of encoder 1209. It should be obvious from the foregoing description that modulo type interleaving can be carried out using any modulo sequence designation up to the size of the interleaver. A modulo-2 interleaver is typically referred to herein as an odd/even interleaver as the modulo sequence designation can have only the values of 1 or 0, i.e., odd or even respectively.

Figure 13 is a graphic illustration of the functioning of a modulo-4 ST interleaver according to an embodiment of the invention. In the illustrated example, the modulo-4 ST interleaver 1301 interleaves a block of 60 tuples. That is the interleaver can accommodate 60 input tuples and perform and interleaving on them. Input tuples 24 through 35 are illustrated at 1303, to demonstrate an exemplary interleaving. Interleaved tuples 0-59 are illustrated at 1305. Input tuples 24 through 35 are illustrated at 1303 as 2 bit tuples. Input tuple 24 includes bit b_{00} which is the LSB or least significant bit of input tuple 24 and b_{01} the MSB or most significant bit of input tuple 24. Similarly, input tuple 25 includes b_{02} which is the least significant bit (LSB) of tuple 25 and b_{03} which is the most significant bit of input tuple 25. Each input tuple 1303 is assigned a modulo sequence designation which is equal to the tuple number modulo-4. The modulo sequence designation of tuple 24 is 0, the modulo sequence designation of tuple 25 is 1, the modulo sequence designation of tuple 26 is 2, the modulo sequence designation of tuple 27 is 3, the modulo sequence designation of tuple 28 is 0 and so forth. Because 1301 is a ST interleaver, the bits of each tuple are interleaved separately. Although the bits of each tuple are interleaved separately, they are interleaved into an interleaved tuple having the same modulo sequence designation, i.e. tuple number mod 4 in the interleaved tuple as in the corresponding input tuple. Accordingly, bit b_{00} the LSB of tuple 24 is interleaved to interleaved tuple number 4 in the least significant bit position. b_{01} the MSB of input tuple 24 is interleaved to interleaved tuple 44 in the most significant bit position. Note that the modulo sequence designation of input tuple 24 is a 0 and modulo sequence designation of interleaved tuple 4 and interleaved tuple 44 are both 0. Accordingly, the criteria that bits of an input tuple having a given modulo sequence designation are interleaved to interleave positions having the same modulo sequence designation. Similarly, b_{02} and b_{03} of input tuple 25 are interleaved to interleaved tuple 57 and interleaved tuple 37 respectively. B_{04} and b_{05} of input tuple 26 are interleaved to interleaved tuples 2 and 22. In like manner the MSB and LSB of all illustrated input tuples 24 through 35 are interleaved to corresponding interleaved tuples having the same modulo sequence designation, as illustrated in Figure 13.

Figure 14A is a graphical illustration of a method for generating an interleaving sequence from a seed interleaving sequence. Interleavers may be implimented in random access memory (RAM). In order to interleave an input sequence, an interleaving sequence may be used. Because interleavers can be quite large, it may be desirable that an interleaving sequence occupy as little storage space within a system as feasible. Therefore, it can be advantageous to generate larger interleaving

sequences from smaller, i.e. seed interleaving sequences. Figure 14A is a portion of a graphical illustration in which a seed interleaving sequence is used to generate four interleaving sequences each the size of the initial seed interleaving sequence. In order to illustrate the generation of sequences from the seed interleaving sequence, an interleaving matrix such as that 1401 may be employed. The interleaving matrix 1401 matches input positions with corresponding output positions. In the interleaving matrix 1401 the input positions I_0 through I_5 are listed sequentially. I_0 is the first interleaving element to enter the interleaving matrix 1401. I_1 is the second element, etc. As will be appreciated by those skilled in the art, the input elements I_0 through I_5 may be considered to be individual bits or tuples. The input positions in the interleaving matrix 1401 are then matched with the seed sequence. By reading through the interleaving matrix 1401 an input position is matched with a corresponding output position. In the illustrative example, of the interleaving matrix 1401, input I_0 is matched with the number 3 of the seed sequence. This means that the I_0 or first element into the interleaving matrix 1401 occupies position 3 in the resulting first sequence. Similarly, I_1 will be matched with a 0 position in sequence 1 and so forth. In other words, the input sequence I_0, I_1, I_2, I_3, I_4, I_5 is reordered according to the seed sequence so that the resulting sequence output from the interleaving matrix 1401 is I_1, I_2, I_5, I_0, I_4, I_3 where the output sequence is obtained by listing the sequence of the output in the usual ascending order I_0, I_1, I_2, I_3, I_4, I_5, where the left most position is the earliest. Put another way, the resulting sequence number 1 is {3, 4, 0, 5, 2, 1}, which corresponds to the subscript of the output sequence 1409. Similarly, in interleaving matrix 1403 also called the inverse interleaving matrix or $INTLV^{-1}$ the input sequence 1400 is accepted by the interleaving matrix 1403 but instead of being written into this interleaving matrix sequentially, as in the case with interleaving matrix 1401, the elements are written into the interleaving matrix according to the seed sequence. The interleaving matrix 1403 is known as the inverse of interleaving matrix 1401 because by applying interleaving matrix 1401 and then successively applying inverse interleaving matrix 1403 to any input sequence, the original sequence is recreated. In other words, the two columns of the interleaving matrix 1401 are swapped in order to get interleaving matrix 1403. Resulting output sequence 1411 is I_3, I_0, I_1, I_5 , I_4, I_2. Therefore, sequence number 2 is equal to 2, 4, 5, 1, 0, 3.

The seed interleaving sequence can also be used to create an additional two sequences. The interleaving matrix 1405 is similar to interleaving matrix 1401 except that the time reversal of the seed sequence is used to map the corresponding output position. The output then of interleaver reverse (INTLVR 1405) is then I_4, I_3, I_0, I_5, I_1, I_2.

Therefore, sequence 3 is equal to 2, 1, 5, 0, 3, 4. Next an interleaving matrix 1407 which is similar to interleaving matrix 1403 is used. Interleaving matrix 1407 has the same input position elements as interleaving matrix 1403, however, except that the time reversal of the inverse of the seed sequence is used for the corresponding output position within interleaving matrix 1407. In such a manner, the input sequence 1400 is reordered to I_2, I_4, I_5, I_1, I_0, I_3. Therefore, sequence number 4 is equal to 3, 0, 1, 5, 4, 2, which are, as previously, the subscripts of the outputs produced. Sequences 1 through 4 have been generated from the seed interleaving sequence. In one embodiment of the invention the seed interleaving sequence is an S random sequence as described by S. Dolinar and D. Divsalar in their paper "Weight Distributions for Turbo Codes Using Random and Non-Random Permeations," TDA progress report 42-121, JPL, August 1995.

Figure 14B is a series of tables illustrating the construction of various modulo interleaving sequences from sequence 1 through 4 (as illustrated in Figure 14A). Table 1 illustrates the first step in creating an interleaving sequence of modulo-2, that is an even/odd interleaving sequence, from sequence 1 and 2 as illustrated in Figure 14A. Sequence 1 is illustrated in row 1 of table 1. Sequence 2 is illustrated in row 2 of table 1. Sequence 1 and sequence 2 are then combined in row 3 of table 1 and are labeled sequence 1-2. In sequence 1-2 elements are selected alternatively, i.e. sequentially from sequence 1 and 2 in order to create sequence 1-2. That is element 1, which is a 1, is selected from sequence 1 and placed as element 1 in sequence 1-2. The first element in sequence 2, which is a 3, is next selected and placed as the second element in sequence 1-2. The next element of sequence 1-2 is selected from sequence 1, the next element is selected from sequence 2, etc. Once sequence 1-2 has been generated, the position of each element in sequence 1-2 is labeled. The position of elements in sequence 1-2 is labeled in row 1 of table 2. The next step in generating the interleaving sequence, which will be sequence 5 is to multiply each of the elements in sequence 1-2 by the modulo of the sequence being created. In this case, we are creating a modulo-2 sequence and therefore, each of the elements in sequence 1-2 will be multiplied by 2. If a modulo-3 sequence had been created in the multiplication step, the elements would be multiplied by 3 as will be seen later. The multiplication step is a step in which the combined sequences are multiplied by the modulo of the interleaving sequence desired to be created.

This methodology can be extended to any modulo desired. Once the sequence 1-2 elements have been multiplied times 2, the values are placed in row 3 of table 2. The next step is to add to each element, now multiplied by modulo-N (here N equals 2)

the modulo-N of the position of the element within the multiplied sequence i.e. the modulo sequence designation. Therefore, in a modulo-2 sequence (such as displayed in table 2) in the 0th position the modulo-2 value of 0 (i.e. a value of 0) is added. To position 1 the modulo-2 value of 1 (i.e. a value of 1) is added, to position 2 the modulo-2 value of 2 (i.e. a value of 0) is added. To position 3 the modulo-2 value of 3 is (i.e. a value of 1) is added. This process continues for every element in the sequence being created. Modulo position number as illustrated in row 4 of table 2 is then added to the modulo multiplied number as illustrated in row 3 of table 2. The result is sequence 5 as illustrated in row five of table 2. Similarly, in table 3, sequence 3 and sequence 4 are interspersed in order to create sequence 3-4. In row 1 of table 4, the position of each element in sequence 3-4 is listed. In row 3 of table 4 each element in the sequence is multiplied by the modulo (in this case 2) of the sequence to be created. Then a modulo of the position number is added to each multiplied element. The result is sequence 6 which is illustrated in row 5 of table 4.

It should be noted that each component sequence in the creation of any modulo interleaver will contain all the same elements as any other component sequence in the creation of a modulo interleaver. Sequence 1 and 2 have the same elements as sequence 3 and 4. Only the order of the elements in the sequence are changed. The order of elements in the component sequence may be changed in any number of a variety of ways. Four sequences have been illustrated as being created through the use of interleaving matrix and a seed sequence, through the use of the inverse interleaving of a seed sequence, through the use of a timed reversed interleaving of a seed sequence and through the use of an inverse of a time interleaved reverse of a seed sequence. The creation of component sequences are not limited to merely the methods illustrated. Multiple other methods of creating randomized and S randomized component sequences are known in the art. As long as the component sequences have the same elements (which are translated into addresses of the interleaving sequence) modulo interleavers can be created from them. The method here described is a method for creating modulo interleavers and not for evaluating the effectiveness of the modulo interleavers. Effectiveness of the modulo interleavers may be dependent on a variety of factors which may be measured in a variety of ways. The subject of the effectiveness of interleavers is one currently of much discussion in the art.

Table 5 is an illustration of the use of sequence 1, 2, and 3 in order to create a modulo-3 interleaving sequence. In row 1 of table 5 sequence 1 is listed. In row 2 of table 5 sequence 2 is listed and in row 3 sequence 3 is listed. The elements of each of the three sequences are then interspersed in row 4 of table 5 to create sequence 1-2-

3.

In table 6 the positions of the elements in sequence 1-2-3 are labeled from 0 to 17. Each value in sequence 1-2-3 is then multiplied by 3, which is the modulo of the interleaving sequence to be created, and the result is placed in row 3 of table 6. In row 4 of table 6 a modulo-3 of each position is listed. The modulo-3 of each position listed will then be added to the sequence in row 3 of table 3, which is the elements of sequence 1-2-3 multiplied by the desired modulo, i.e. 3. Sequence 7 is then the result of adding the sequence 1-2-3 multiplied by 3 and adding the modulo-3 of the position of each element in sequence 1-2-3. The resulting sequence 7 is illustrated in table 7 at row 5. As can be seen, sequence 7 is a sequence of elements in which the element in the 0 position mod 3 is 0. The element in position 1 mod 3 is 1. The element in position 2 mod 3 is 2. The element in position 3 mod 3 is 0 and so forth. This confirms the fact that sequence 7 is a modulo-3 interleaving sequence. Similarly, sequence 5 and 6 can be confirmed as modulo-2 interleaving sequences by noting the fact that each element in sequence 5 and sequence 6 is an alternating even and odd (i.e. modulo-2 equals 0 or modulo-2 equals 1) element.

Figure 14C is a graphical illustration of creating a modulo-4 sequence from four component sequences. In table 7 sequences 1, 2, 3 and 4 from Figure 14A are listed. The elements of sequence 1, 2, 3 and 4 are then interspersed to form sequence 1-2-3-4.

In table 8 row 1 the positions of each element in sequence 1-2-3-4 are listed. In row 3 of table 8 each element of sequence 1-2-3-4 is multiplied by a 4 as it is desired to create a modulo-4 interleaving sequence. Once the elements of sequence 1-2-3-4 have been multiplied by 4 as illustrated in row 3 of table 8, each element has added to it a modulo-4 of the position number, i.e. the modulo sequence designation of that element within the 1-2-3-4 sequence. The multiplied value of sequence 1-2-3-4 is then added to the modulo-4 of the position in sequence 8 results. Sequence 8 is listed in row 5 of table 8. To verify that the sequence 8 generated is a modulo-4 interleaving sequence each number in the sequence can be divided mod 4. When each element in sequence 6 is divided modulo-4 sequence of 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3 etc. results. Thus, it is confirmed that sequence 8 is a modulo-4 interleaving sequence, which can be used to take an input sequence of tuples and create a modulo interleaved sequence of tuples.

Figure 15 is a general graphical illustration of trellis-coded modulation (TCM). In Figure 15, input tuples designated 1501 are coupled into a trellis encoder 1503. Input tuples, for illustration purposes are designated T_0, T_1, T_2 and T_3. Within the trellis

encoder 1503 the input tuples 1501 are accepted by a convolutional encoder 1505. The input tuples that have been convolutionally encoded are mapped in a mapper 1507. The TCM process yields a signal constellation represented as a set of amplitude phase points (or vectors) on an In phase Quadrature (I-Q) plane. An example of such vectors illustrated at 1509, 1511, 1513, and 1515. The vector represented in the I-Q (In phase and Quadrature) illustration is well known in the art. The process of convolutionally encoding and mapping when taken together is generally referred to as trellis-coded modulation. A similar process called turbo trellis-coded modulation (TTCM) is illustrated in Figure 16.

Figure 16 is a graphical illustration of TTCM (Turbo Trellis Coded Modulation) encoding. In Figure 16 input tuples 1601 are provided to a parallel concatenated (turbo) encoding module 1603. The parallel concatenated turbo encoding module 1603 may comprise a number of encoders and interleavers. Alternatively, the parallel concatenated encoder 1603 may comprise a minimum of two encoders and one interleaver. The output of the turbo encoder is then provided to an output selection and puncturing module. In module 1605 outputs are selected from the constituent encoders of the module 1603. The selection of outputs of the different encoders is sometimes termed puncturing by various sources in the art, because some of the code bits (or parity bits) may be eliminated). Selection of outputs of the constituent encoders within the present disclosure will be referred to herein as selecting. The term selecting is used because, in embodiments of the present invention, encoded tuples are selected from different encoders, but encoded tuples corresponding to each of the input tuples are represented. For example, there may be an encoder designated the odd encoder from which tuples corresponding to encoded versions of odd input tuples are selected. The other encoder may be termed an even encoder in which the coded versions of the even tuples are selected. This process is termed selecting because even though alternating encoded tuples are selected from different encoders a coded version of each input is represented. That is, in the selection process though some encoded symbols are discarded from one encoder and some encoded symbols are discarded from other constituent encoder(s) the selection and modulo interleaving process is such that encoded versions of all input elements are represented. By modulo encoding and selecting sequentially from all encoders, encoded versions of all input bits are represented. The term puncturing as used herein will be used to describe discarding parts or all of encoded tuples which have already been selected. The selected tuples are provided to a mapping 1607. In embodiments of the present invention the mapping may be dependent on the source of the tuple being mapped. That is, the mapping may

be changed for example depending on whether the tuple being mapped has been encoded or not. For example, a tuple from one of the encoders may be mapped in a first mapping. An uncoded tuple which has bypassed the encoder however may be mapped in a second mapping. Combination tuples in which part of the tuple is encoded and part of it is uncoded may also have different mappings. A combination of 3 blocks - block 1603, parallel concatenated encoding, block 1605, output selection and puncturing, and block 1607 mapping comprise what is known as the turbo trellis-coded modulation (TTCM) encoder 1609. The output of the TTCM encoder is a series of constellation vectors as illustrated by examples at 1611, 1613, 1615 and 1617.

Figure 17 is a graphical illustration of a rate 2/3 encoder according to an embodiment of the invention. In Figure 17, input tuples T_0 and T_1 represented at 1701 are provided to odd encoder 1703. Tuple T_0 comprises bits, b_0 and b_1 tuple T_1 comprises bits b_2 and b_3. The input tuples T_0 and T_1 are also provided to an interleaver 1705. Interleaver 1705 accepts input tuples (such as T_0 and T_1) and after interleaving, provides the interleaved tuples to the even encoder 1709. When odd encoder 1703 is accepting tuple T_0, comprising bits b_0 and b_1 , even encoder 1709 is accepting an interleaved tuple comprising bits i_0, and i_1. Similarly, when odd encoder 1703 is accepting tuple T_1 comprising bits b_2 and b_3 even encoder 1709 is accepting an interleaved tuple comprising bits i_2 and i_3. At each encoder clock (EC) both encoders accept an input tuple. The interleaver 1703 is a modulo-2 (even/odd) ST interleaver. Each encoder accepts every input tuple. The even/odd designation refers to which encoded tuple is selected to be accepted by the mapper 1715. By maintaining an even/odd interleaving sequence and by selecting encoded tuples alternatively from one then the other encoder, it can be assured that an encoded version of every input tuple is selected and passed on to the mapper 1715. For example, the encoded tuple 1711, comprising bits c_3 and c_4, and c_5 and corresponding to tuple T_1 is selected and passed onto mapper 1715, which maps both even and odd selections according to map 0.

The encoded tuple c_0, c_1 and c_2, corresponding to input tuple T_0 is not selected from the odd encoder 1703. Instead, the tuple comprising bits c'_0, c'_1, and c'_0, which corresponds to the interleaved input i_0 and i_1 is selected and passed on to mapper 1715, where it is mapped using map 0.

Accordingly, all the components of each tuple are encoded in the odd encoder and all components of each tuple are also encoded in the even encoder. However, only encoded tuples corresponding to input tuples having an odd modulo sequence designation are selected from odd encoder 1703 and passed to the mapper 1715. Similarly only encoded tuples corresponding to input tuples having an even modulo

sequence designation are selected from even encoder 1709 and passed to mapper 1703. Therefore, the odd and even designation of the encoders designate which tuples are selected from that encoder for the purposes of being mapped.

Both encoder 1703 and 1709 in the present example of Figure 17 are convolutional, nonsystematic, recursive encoders according to Figure 5. Although only encoded versions of odd tuples are selected from encoder 1703, and only encoded versions of even tuples are selected from encoder 1709, because both encoders have memory, each encoded output tuple not only contains information from the tuple encoded, but also from previous tuples.

The even/odd encoder of Figure 17 could be modified by including modulo-N interleaving, modulo-N interleaving could be accomplished by adding the appropriate number of both interleavers and encoders, to form a modulo-N TTCM encoder. Additionally, other configurations may be possible. For example, interleaver 1705 may be a ST interleaver. As an alternate another interleaver may be added prior to odd encoder 1703. For example, if a bit interleaver, to separate the input tuple bits were added prior to encoder 1703, and interleaver 1705 were an IT interleaver, the overall effect would be similar to specifying interleaver 1705 to be an ST interleaver.

Both encoders 1703 and 1709 are rate 2/3 encoders. They are both nonsystematic convolutional recursive encoders but are not be limited to such.

The overall TTCM encoder is a 2/3 encoder because both the odd encoder 1703 and the even encoder 1709 accept an input tuple comprising 2 bits and output an encoded output tuple comprising 3 bits. So even though the output to mapper 0 switches between even and odd encoders, both encoders are rate 2/3 and the overall rate of the TTCM encoder of Figure 17 remains at 2/3.

Figure 18 is a graphical illustration of a rate ½ TTCM encoder implimented using the constituent rate 2/3 base encoders, according to an embodiment of the invention. In Figure 18, exemplary input tuples T_0 and T_1 are provided to the TTCM encoder 1800. The T_0 tuple comprises a single bit b_0 and the T_1 tuple comprises a single bit b_1. b_0 and b_1 corresponding to tuples T_0 and T_1 are provided to odd encoder 1803. Both b_0 and b_1 are also provided to interleaver 1805. At the time when odd encoder 1803 is accepting b_0 even encoder is accepting i_0. i_0 is an output of the interleaver 1805. Similarly, i_1 is a output of interleaver 1805 that is provided to even encoder 1809 at the same time that bit b_1 is provided to odd encoder 1803. The interleaver 1805 is an odd/even interleaver (modulo-2). In such a manner when an odd tuple is being provided to odd encoder 1803, an interleaver odd tuple is being provided to even encoder 1809. When an even tuple is being provided to odd 1803, an even interleaved

tuple is being provided to even encoder 1809. In order to achieve a rate ½ code from rate 2/3 constituent encoders, in addition to an input comprising a single input bit, a constant bit value provided to 1811 is a second input of each of the constituent rate 2/3 encoders 1803 and 1809. In Figure 18A the input bit is shown as being a 0 but could just as easily be set to a constant value of 1. Additionally, each encoder input bit might be inputted twice to the odd encoder 1803 and the even encoder 1809 as illustrated in Figure 18B. Multiple other configurations are possible. For example both encoders might receive both input tuples as illustrated in Figure 18C, or one of the inputs might be inverted as in Figure 18E. Additionally hybrid combinations, such as illustrated in Figure 18D are possible.

The output of odd encoder 1803, which corresponds to input tuple T_0, comprises bits c_0, c_1, c_2. The output tuple of odd encoder 1803 corresponding to tuple T_1 comprises bits c_3, c_4, and c_5. At encoder clock EC_0 the even encoder 1809 has produced an encoded output tuple having bits c'_0, c'_1 and c'_2. One of the three encoded bits, in the present illustration c'_2, is punctured i.e. dropped and the remaining 2 bits are then passed through to mapper 1813. During the odd encoder clock OC_1 two of three of the encoded bits provided by odd encoder 1803 are selected and passed to mapper 1813. Output bit c_4 is illustrated as punctured, that is being dropped and not being passed through the output mapper 1813. Mapper 1813 employs map number 3 illustrated further in Figure 24. For each encoder clock a single input tuple comprising 1 bit is accepted into the TTCM encoder 1800. For each clock a 2-bit encoded quantity is accepted by mapper 1813. Because for each one bit provided to the encoder, 2 bits are outputted, therefore the encoder is a rate ½ encoder. The odd and even encoders in the present embodiment are nonsystematic, convolutional, recursive encoders, but are not limited to such. The encoders may be any combination, for example such as systematic, block encoders. Interleaver 1805 is an odd/even interleaver and so odd output tuples are accepted by the mapper 1813 from odd encoder 1803 and even encoded tuples are accepted by the mapper 1813 from even encoder 1809. In such a manner, all input tuples are represented in the output accepted by mapper 1813, even though some of the redundancy is punctured. Mapper 1813 utilizes map 3 as illustrated in Figure 25 for use by rate ½ TTCM encoder 1800.

Figure 19 is a graphical illustration of a rate 3/4 TTCM encoder, having constituent 2/3 rate encoders, according to an embodiment of the invention. In Figure 19 the input tuples T_0 and T_1, illustrated at 1901, comprise 3 bit input tuples. Input tuple T_0 comprises bits b_0, b_1 and b_2. Input tuple T_1 comprises bits b_3, b_4 and b_5. Bit $\underline{b_2}$ of input tuple T_0 is underlined as is $\underline{b_5}$ of input tuple T_1. Bits $\underline{b_2}$ and $\underline{b_5}$ are underlined

because neither of these bits will pass through either encoder. Instead, these bits will be concatenated to the output of the even or odd encoder and the resulting in a 4 bit tuple provided to mapper 1911. b_0 and b_1 of input tuple T_0 are provided to odd encoder 1903. At the same time that b_0 and b_1 are being accepted by the odd encoder 1903, interleaved bits i_0 and i_1 are being accepted by even encoder 1909. Interleaver 1905 is an odd/even (module-2) type interleaver. The encoders illustrated at 1903 and 1909 are the encoders illustrated in Figure 5. Encoders 1903 and 1909 are the same as the encoders illustrated at 1803 and 1809 in Figure 18, 1703 and 1709 in Figure 17 and as will be illustrated at 2003 and 2009 in Figure 20A and 2103 and 2109 in Figure 21A In other words, the odd encoder and even encoder are rate 2/3, nonsystematic, convolutional recursive encoders. Other types of encoders may however be used, and types may be mixed and matched as desired.

Figure 17 through 21 are encoding arrangements that utilize the same basic encoder as illustrated in Figure 5. In Figure 19, encoders 1903 and 1909 are illustrated as separate encoders for conceptual purposes. Those skilled in the art will realize that a single encoder may be used and may be time-shared. Figures 17 through 21 are conceptual type Figures and are figures that represent general concepts. They depict the general concept accurately regardless of the particular implementation of circuitry chosen. In the rate 3/4 encoder of Figure 19, the input tuples T_0, T_1 (and all other input tuples to the encoder of Figure 19) comprise 3 bits. Since encoders 1903 and 1909 are rate 2/3 encoders with 2 input bits, then only 2 bits can be accommodated at a particular time. Accordingly, bit $\underline{b_2}$ of tuple T_0 and bit $\underline{b_5}$ of tuples T_1 bypass the encoders completely. $\underline{b_5}$ is concatenated to the output of odd encoder 1903, i.e. c_3, c_4 and c_5 the combination of encoder tuple c_3, c_4, c_5 and $\underline{b_5}$ are then provided to mapper 1911 which maps the output according to map 2. Map 2 is illustrated in Figure 24. Similarly, the output of even encoder 1909, comprising encoded bits c'_0, c'_1 and c'_2, is combined with bit $\underline{b_2}$ of input tuple T_0 and then the combination of $\underline{b_2}$, c'_0, c'_1, c'_2 is provided to mapper 1911. In such a way the three bits of encoded tuples are converted into four bits for mapping in mapper 1911. The four bits mapped comprise the three encoded bits from either the odd or even encoder plus a bit from the input tuple which has by passed both encoders.

Figure 20A is a graphical illustration of a rate 5/6 TTCM encoder, having constituent 2/3 rate basic encoders, according to an embodiment of the invention. In Figure 20A the input tuples T_0 and T_1 are illustrated at 2001. Input tuple T_0 comprises five bits, b_0 through b_4. Input tuple T_1 also comprises five bits, b_5 through b_9. $\underline{b_4}$ of tuple T_0 and $\underline{b_9}$ of tuple T_1 are underlined to illustrate that they do not pass through either

encoder. The odd encoder 2003 accepts b_0 and b_1 during a first encoder clock time during which even encoder 2009 is accepting interleaved bits i_0 and i_1. Bits i_0 and i_1 are the outputs of the interleaver 2005 that correspond to the same time during which inputs b_0 and b_1 are accepted from the odd encoder. Similarly, the odd encoder 2003 is accepting bits b_2 and b_3 at a time when the even encoder 2009 is accepting bits i_2 and i_3. Similarly, input tuple T_1, is separated into 2 bit encoder input tuples because the constituent encoders are rate 2/3 encoders which accept 2 bits input and produce three encoded bits out. Because each input tuple 2001 is five bits and because each encoder allows only a 2 bit input, input tuple T_0 is separated into encoder tuple b_0 and b_1 and encoder tuple b_2 and b_3. The encoder therefore, must process two encoder input tuples for each input tuple 2001. Therefore, a single input tuple 2001 will require two encoder clocks for processing. The even encoder 2009 encodes tuple i_0 and i_1 and produces corresponding output code bits c'_0, c'_1 and c'_2. After processing i_0 and i_1 the even encoder 2009 processes i_2 and i_3 The output of even encoder 2009, which corresponds to input bits i_2 and i_3 is c'_3, c'_4 and c'_5. The odd encoder 2003 processes a first tuple b_0 and b_1 and then processes a second tuple b_2 and b_3. Tuple b_0 and b_1 are accepted by encoder 2003 which produces a corresponding encoded 3 bit tuple c_0, c_1 and c_2. After accepting b_0 and b_1, the odd encoder 2003 accepts second tuple b_2 and b_3 and produces a corresponding output c_3, c_4, and c_5. Encoder output c'_0, c'_1 and c'_2 corresponding to encoder tuple i_1 and i_0 are provided to mapper 2011. Mapper 2011 uses map 0 to map c'_0, c'_1 and c'_2. Subsequently to producing c'_0, c'_1 and c'_2 even encoder 2009 accepts i_2 and i_3 and produces output c_3, c_4, and c_5. Instead of selecting c_3, c_4, c_5 to be mapped, uncoded bit $\underline{b}_4$ is combined with interleaved bits i_2 and i_3 and selected. i_2, i_3 and $\underline{b}_4$ are then accepted by mapper 2011, which employs map 1 to map bits i_2, i_3 and $\underline{b}_4$. Therefore, with respect to the overall input tuple T_0 five bits are input into the TTCM encoder 2000 and six bits are passed to mapper 2011. In other words, a coding rate of 5/6 is generated. Similarly, odd encoder 2003 encodes bits b_5 and b_6 and produces coded bits c_6, c_7 and c_8. Subsequently odd encoder 2003 encodes bits b_7 and b_8 and produces coded bits c_9, c_{10} and c_{11}. c_6, c_7 and c_8 are passed to the encoder 2001 as is where they are mapped using map 0. Encoded bit c_9, c_{10} and c_{11}, however, are punctured, i.e. they are dropped and instead bits b_7, b_8 and $\underline{b}_9$ are substituted. b_7, b_8 and $\underline{b}_9$ are passed to encoder 2011 which uses map 1 to map b_7, b_8, and $\underline{b}_9$. A graphical illustration of map 0 can be found in Figure 22 and a graphical illustration of Map 1 can be found in Figure 23. In the manner just described, a rate 5/6 TTCM encoder is realized from two component rate 2/3 encoders. Interleaver 2005 is similar to interleaver 1705, 1805, 1905, 2005 and 2105 which also are even/odd or

modulo-2 type interleavers. Other modulo interleavers, just as with all other embodiments illustrated in figures 17 through 21, can be realized by adding additional interleavers and encoders and by selecting outputs and uncoded bits in a straight format manner similar to that illustrated in Figure 20A.

Figure 20B represents an alternate encoding that will yield the same coding rate as Figure 20A.

Figure 21A is a graphical illustration of a rate 8/9 TTCM encoder realized using constituent rate 2/3 encoder, according to an embodiment of the invention. To illustrate the functioning of TTCM rate 8/9 encoder 2100 two sequential input tuples T_0 and T_1, illustrated at 2101, will be considered. Since the constituent encoders are rate 2/3 having two bits as input and three bits as output, the input tuples will have to be subdivided into encoder tuples. In other words, the input tuples will be divided into tuple pairs which can be accepted by odd encoder 2103 and even encoder 2109. Odd encoder 2103 accepts tuple pair b_0 and b_1, pair b_2 and b_3, pair b_4 and b_5, pair b_8 and b_9, pair b_{10} and b_{11}, and pair b_{12} and b_{13} sequentially, since the basic 2/3 rate encoder can only accept one pair of input bits at a time. Even encoder correspondingly accepts input pairs i_0 and i_1, input pair i_2 and i_3, input pair i_4 and i_5, input pair i_8 and i_9, input pair i_{10} and i_{11}, and input pair i_{12} and i_{13} sequentially. The pairs accepted by the even encoder correspond to tuple pairs having the same numbering accepted by the odd encoder at the same time. That is i_0 and i_1 are accepted by the even encoder 2109 during the same time period as input pair b_0 and b_1 is accepted by the odd encoder 2103. Odd and even encoders then produce encoded outputs from the input pairs accepted. Even encoder 2909 produces a first encoded output triplet c'_0, c'_1 and c'_2 followed by a second output triplet c'_3, c'_4 and c'_5 followed by a third output triplet c'_6, c'_7 and c'_8 (a triplet is a 3 bit tuple). The first output triplet c'_0, c'_1 and c'_2 is accepted by the mapper 2111. The mapper 2111 utilizes map 0 to map encoded output c'_0, c'_1 and c'_2. Encoded output bits c'_3, c'_4 and c'_5 however are punctured, that is not sent to the mapper. Instead of sending c'_3, c'_4 and c'_5 to the mapper 2111 the triplet of bits comprising i_2, i_3 and $\underline{b}_6$ are sent to the mapper 2111. The mapper 2111 utilizes map 1 as the mapping for the triplet i_2, i_3, $\underline{b}_6$. Encoded triplet c'_6, c'_7 and c'_8 is also punctured. That is, it is not sent to the mapper 2111. Instead, i_4, i_5 and $\underline{b}_7$ is sent to the mapper 2111 which uses map 1 to map input triplet i_4, i_5 and $\underline{b}_7$. Because eight bits corresponding to tuple T_0 are accepted by the even encoder 2109 and nine bits are output into the mapper 2111 the overall encoder 2100 is a rate 8/9 encoder. Similarly, input tuple T_1 is encoded by the odd encoder 2103. The output triplet from the odd encoder c_9, c_{10} and c_{11} corresponds to input tuple b_8 and b_9. Next, odd encoder 2103

produces an encoded output triplet c_{12}, c_{13} and c_{14}, which is an output triplet corresponding to input pair b_{10} and b_{11}. Subsequently odd encoder 2103 produces output triplet c_{15}, c_{16} and c_{17}. Output triplet c_{15}, c_{16} and c_{17} corresponds to input pair b_{12} and b_{13}. Output triplet c_9, c_{10} and c_{11} are sent to the mapper 2111 which uses map 0 to map output triplet c_9, c_{10} and c_{11}. Output triplet c_{12}, c_{13} and c_{14} however is punctured and in its place b_{10}, b_{11} and $\underline{b}_{14}$ is sent to mapper 2111 where map 1 is employed to map the input triplet b_{10}, b_{11} and $\underline{b}_{14}$. The encoder triplet c_{15}, c_{16} and c_{17} is also punctured and a triplet comprising b_{12}, b_{13} and $\underline{b}_{15}$ is provided to mapper 2111. Map 1 is used to map the input triplet b_{12}, b_{13} and $\underline{b}_{15}$. In the manner just described an 8/9 encoder is fabricated from two constituent rate 2/3 encoder.

From the foregoing TTCM encoder examples of Figures 17 through 21 it is seen that the basic rate 2/3 encoders can be used in a variety of configurations to produce a variety of coding rates.

The basic constituent encoders illustrated in Figures 17 through 21 are rate 2/3, nonsystematic, convolutional recursive encoders. These illustrations represent a few examples. Different types of encoders and even different rates of encoders may yield many other similar examples. Additionally, encoder types can be mixed and matched; for example, a recursive nonsystematic convolution encoder may be used with a nonrecursive systematic block encoder.

Additionally, the interleavers illustrated in Figures 17 through 21 are modulo-2 (even/odd) ST interleavers. Those skilled in the art will realize that IT type interleavers may be used alternatively in the embodiments of the invention illustrated in Figures 17 through 21.

Additionally the TTCM encoders illustrated in Figures 17 through 21 may employ modulo-N encoding systems instead of the modulo-2 (even/odd) encoding systems illustrated. For example, each of the constituent encoder - modulo-2 interleaver subsystems may be replaced by modulo-N subsystems such as illustrated in Figure 8A. By maintaining the same type puncturing and selecting with each encoder as displayed with the even/odd encoders of Figures 17 through 21 and extending it to modulo-N systems, such as illustrated in Figure 8A, the same coding rates can be maintained in a modulo-N system for any desired value N.

Figure 21B represents an alternate encoding that will yield the same coding rate as Figure 21A

Figure 22 is a graphical illustration of map 0 according to an embodiment of the invention. Map 0 is used in the implementation of the rate 2/3 encoder as illustrated in Figure 17. Map 0 is also utilized in rate 5/6 encoder illustrating in Figure 20A and rate

8/9 encoder illustrated in Figure 21A.

Figure 23 is a graphical illustration of map 1 according to an embodiment of the invention. Map 1 is used by the mapper in the rate 5/6 encoder in Figure 20A, and in the mapper in the rate 8/9 encoder in Figure 21A.

Figure 24 is a graphical illustration of map 2 according to an embodiment of the invention. Map 2 is utilized in the fabrication of the rate 3/4 encoder as illustrated in Figure 19.

Figure 25 is a graphical illustration of map 3 according to an embodiment of the invention. Map 3 is used in the rate ½ encoder as depicted in Figure 18.

Maps 0 through 3 are chosen through a process different from the traditional approach of performing an Ungerboeck mapping (as given in the classic work "Channel Coding with Multilevel/Phase Signals" by Gottfried Ungerboeck, IEEE Transactions on Information Theory Vol. 28 No. 1 January 1982). In contrast in embodiments of the present invention, the approach used to develop the mappings was to select non Ungerboeck mappings, then to measure the distance between the code words of the mapping. Mappings with the greatest average effective distance are selected. Finally the mappings with the greatest average effective distance are simulated and those with the best performance are selected. Average effective distance is as described by S. Dolinar and D. Divsalar in their paper "Weight Distributions for Turbo Codes Using Random and Non-Random Permeations," TDA progress report 42-121, JPL, August 1995.

Figure 26 is a TTCM decoder according to an embodiment of the invention. Figure 26 illustrates a block diagram of the TTCM decoder corresponding to the TTCM encoder described above. The TTCM decoder includes a circular buffer 2602, a metric calculator module 2604, two soft-in soft-out (SISO) modules 2606, 2608, two interleavers 2610, 2612, a conditional points processing module 2614, a first-in first-out (FIFO) register 2616, and an output processor 2618.

The TTCM decoder of Figure 26 impliments a MAP (Maximum A Posteriori) probability decoding algorithm.

The MAP Algorithm is used to determine the likelihood of the possible particular information bits transmitted at a particular bit time.

Turbo decoders, in general, may employ a SOVA (Soft Output Viterbi Algorithm) for decoding. SOVA is derived from the classical Viterbi Decoding Algorithm (VDA). The classical VDA takes soft inputs and produces hard outputs a sequence of ones and zeros. The hard outputs are estimates of values, of a sequence of information bits. In general, the SOVA Algorithm takes the hard outputs of the classical VDA and produces

weightings that represent the reliability of the hard outputs.

The MAP Algorithm, implimented in the TTCM decoder of Figure 26, does not produce an intermediate hard output representing the estimated values of a sequence of transmitted information bits. The MAP Algorithm receives soft inputs and produces soft outputs directly.

The input to the circular buffer i.e. input queue 2602 is a sequence of received tuples. In the embodiments of the invention illustrated in Figure 26, each of the tuples is in the form of 8-bit in-phase (I) and 8-bit quadrature (Q) signal sample where each sample represents a received signal point or vector in the I-Q plane. The circular buffer 2602 outputs one tuple at a time to the metric calculator 2604.

The metric calculator 2604 receives I and Q values from the circular buffer 2602 and computes corresponding metrics representing distances form each of the 8 members of the signal constellation (using a designated MAP) to the received signal sample. The metric calculator 2604 then provides all eight distance metrics (soft inputs) to the SISO modules 2606 and 2608. The distance metric of a received sample point from each of the constellation points represents the log likelihood probability that the received sample corresponds to a particular constellation point. For rate 2/3, there are 8 metrics corresponding to the points in the constellation of whatever map is used to encode the data. In this case, the 8 metrics are equivalent to the Euclidean square distances between the value received and each of the constellation whatever map is used to encode the data..

SISO modules 2606 and 2608 are MAP type decoders that receive metrics from the metric calculator 2604. The SISOs then perform computations on the metrics and pass the resulting A Posteriori Probability (APoP) values or functions thereof (soft values) to the output processor 2618.

The decoding process is done in iterations. The SISO module 2606 decodes the soft values which are metrics of the received values of the first constituent code corresponding to the constituent encoder for example 1703 (Figure 17). The SISO module 2608 decodes the soft values which are the APoP metrics of the received values of the second constituent code corresponding to the constituent encoder for example 1709 (Figure 17). The SISO modules simultaneously process both codes in parallel. Each of the SISO modules computes the metrics corresponding to the input bits for every bit position of the in the block of 10K tuples (representing a exemplary block of date), and for each of the trellis states that the corresponding encoder could have been in.

One feature of the TTCM decoder is that, during each iteration, the two SISO

modules 2606, 2608 are operating in parallel. At the conclusion of each iteration, output from each SISO module is passed through a corresponding interleaver and the output of the interleaver is provided as updated or refined A Priori Probability (APrP) information to the input of other cross coupled SISO modules for the next iteration.

After the first iteration, the SISO modules 2706, 2708 produce soft outputs to the interleaver 2610 and inverse interleaver 2612, respectively. The interleaver 2610 (respectively, inverse interleaver 2612) interleaves the output from the SISO module 2606 (respectively, 2608) and provides the resulting value to the SISO module 2608 (respectively, 2606) as a priori information for the next iteration. Each of the SISO modules use both the metrics from the metric calculator 2604 and the updated APrP metric information from the other cross coupled SISO to produce a further SISO Iteration. In the present embodiment of the invention, the TTCM decoder uses 8 iterations in its decoding cycle. The number of iterations can be adjusted in firmware or can be changed depending on the decoding process.

Because the component decoders SISO 2606 and 2608 operate in parallel, and because the SISO decoders are cross coupled, no additional decoders need to be used regardless of the number of iterations made. The parallel cross coupled decoders can perform any number of decoding cycles using the same parallel cross coupled SISO units (e.g. 2606 and 2608).

At the end of the 8 iterations the iteratively processed APoP metrics are passed to the output processor 2618. For code rate 2/3, the output processor 2618 uses the APoP metrics output from the interleaver 2610 and the inverse interleaver 2612 to determine the 2 information bits of the transmitted tuple. For code rate 5/6 or 8/9, the output from the FIFO 2616, which is the delayed output of the conditional points processing module 2614, is additionally needed by the output processor 2618 to determine the uncoded bit, if one is present.

For rate 2/3, the conditional points processing module 2614 is not needed because there is no uncoded bit. For rate 5/6 or 8/9, the conditional points processing module 2614 determines which points of the received constellation represent the uncoded bits. The output processor 2618 uses the output of the SISOs and the output of the conditional points processor 2614 to determine the value of the uncoded bit(s) that was sent by the turbo-trellis encoder. Such methodology of determining the value of an uncoded bit(s) is well known in the art as applied to trellis coding.

Figure 27 is a TTCM modulo-4 decoder according to an embodiment of the invention. The modulo four decoder of Figure 27 is similar to the modulo-2 decoder illustration in Figure 26. The functions of the input queue 2802, metric calculator 2804,

conditional points processor 2814, and first in first out (FIFO) 2816 are similar to their counterparts in Figure 26. The signals that will be decoded by the TTCM modulo-4 decoder Figure 27 is one that has been coded in a modulo-4 interleaving system. Therefore, instead of having merely even and odd SISOs and interleavers, SISO 0, 1, 2 and 3 are used as are interleaver 0, 1, 2 and 3. Because the data has been encoded using a modulo-4 interleaving system, SISOs 0, 1, 2 and 3 can operate in parallel using interleaver 0, 1, 2 and 3. Once the SISOs 0 through 3 have processed through the points corresponding to the metrics of the points received in the input queue, the points can then be passed on to output process 2818. Output process 2818 will then provide decoded tuples.

Figure 28 is a graphical illustration of a modulo-N and encoding and decoding system according to an embodiment of the invention. In Figure 28, the encoder 2800 is a modulo-N encoder. The modulo-N encoder illustrated has N encoders and N-1 interleavers. The selector, 2801 selects encoded tuples sequentially from the output of encoders 0 through N. Selector 2801 then passes the selection onto the mapper which applies the appropriate mapping. The appropriately mapped data is then communicated over a channel 2803 to an input queue 2805. The functions of input 2805, metric calculator 2807, conditional points processor 2809 and FIFO 2811 are similar to those illustrated in Figures 26 and 2478. The decoder 2813 has N SISOs corresponding to the N encoders. Any desired amount of parallelism can be selected for the encoder decoder system with the one caveat that the modulo-N decoding must match the modulo-N encoding. By increasing the modulo of the system, more points which have been produced by the metric calculator 2807 can be processed at the same time.

SISOs 0 through N process the points provided by the metric calculator in parallel. The output of one SISO provides A Priori values for the next SISO. For example SISO 0 will provide an A Priori value for SISO 1, SISO 1 will provide an A Priori value for SISO 2, etc. This is made possible because SISO 0 impliments a Map decoding algorithm and processes points that have a modulo sequence position of 0 within the block of data being processed, SISO 1 impliments a Map decoding algorithm and processes points that have a modulo sequence position of 1 within the block of data being processed, and so forth. By matching the modulo of the encoding system to the modulo of the decoding system the decoding of the data transmitted can be done in parallel. The amount of parallel processing available is limited only by the size of the data block being processed and the modulo of the encoding and decoding system that can be implimented.

WHAT IS CLAIMED IS:

1. A method for encoding, to provide a quasi error free decoding, the method comprising:

providing data to be communicated;

algebraically encoding the data to be communicated thereby producing algebraically encoded data;

interleaving the algebraic encoded data in an interleaver having a guaranteed Depth thereby producing interleaved data; and

turbo encoding the interleaved data thereby producing turbo encoded data;

2. The method of claim 1 wherein algebraically encoding the data to be communicated comprises Reed-Solomon encoding the data to be communicated.

3. The method of claim 1 wherein interleaving the algebraic encoded data comprises using a Ramsey interleaver to interleave the algebraic encoded data.

4. The method of claim 1 wherein turbo encoding the interleaved data comprises modulo-N turbo encoding of the interleaved data.

5. The method of claim 4 wherein N>2.

6. The method of claim 1 wherein turbo encoding the interleaved data further comprises providing at least one nonsystematic component encoding.

7. The method of claim 1 wherein the guaranteed Depth is at least 201.

8. A method for decoding quasi error free encoded data the method comprising:

accepting data to be decoded which has been encoded according to the method of claim 1;

turbo decoding the data to be decoded;

deinterleaving the data using a deinterleaver having a guaranteed depth to produce deinterleaved data; and

algebraically decoding the deinterleaved data to produce decoded data.

9. The method of claim 8 wherein decoding the data to be communicated

comprises Reed-Solomon decoding the data to be communicated.

10. The method of claim 8 wherein deinterleaving the data to be decoded comprises using a Ramsey deinterleaver to deinterleave the algebraic encoded data.

11. The method of claim 8 wherein turbo decoding the interleaved data comprises modulo-N turbo decoding of the interleaved data.

12. The method of claim 8 wherein N>2.

13. The method of claim 11 wherein modulo-N turbo decoding further comprises parallel map decoding.

14. An encoding apparatus the apparatus comprising:

an input that accepts data to be encoded;

an algebraic encoder that receives the data to be encoded and algebraically encodes the data thereby producing algebraically encoded data;

an interleaver that interleaves the algebraically encoded data, said interleaver having a guaranteed Depth, thereby producing interleaved data; and

a turbo encoder that encodes the interleaved data thereby producing turbo encoded data;

15. The apparatus of claim 14 wherein the algebraic encoder comprises a Reed-Solomon encoder.

16. The apparatus of claim 14 wherein the interleaver comprises a Ramsey interleaver.

17. The apparatus of claim 14 wherein the turbo encoder comprises a modulo-N turbo encoder.

18. The apparatus of claim 4 wherein N>2.

19. The apparatus of claim 14 wherein the turbo encoder comprises a turbo-trellis encoder.

20. A decoder apparatus comprising:

a turbo decoder that accepts data to be decoded, which has been encoded according to the method of claim 1, to produce turbo decoded data;

a deinterleaver that accepts the turbo decoded data and produce deinterleaved data; and

an algebraic decoder that decodes the deinterleaved data to produce decoded data.

21. The apparatus of claim 20 wherein the algebraic decoder comprises a Reed-Solomon decoder.

22. The apparatus of claim 20 wherein the deinterleaver comprises a Ramsey deinterleaver .

23. The apparatus of claim 20 wherein the turbo decoder comprises a modulo-N turbo decoder.

24. The apparatus of claim 20 wherein N>2.

25. The method of claim 11 wherein the modulo-N turbo decoder is a parallel map decoder.

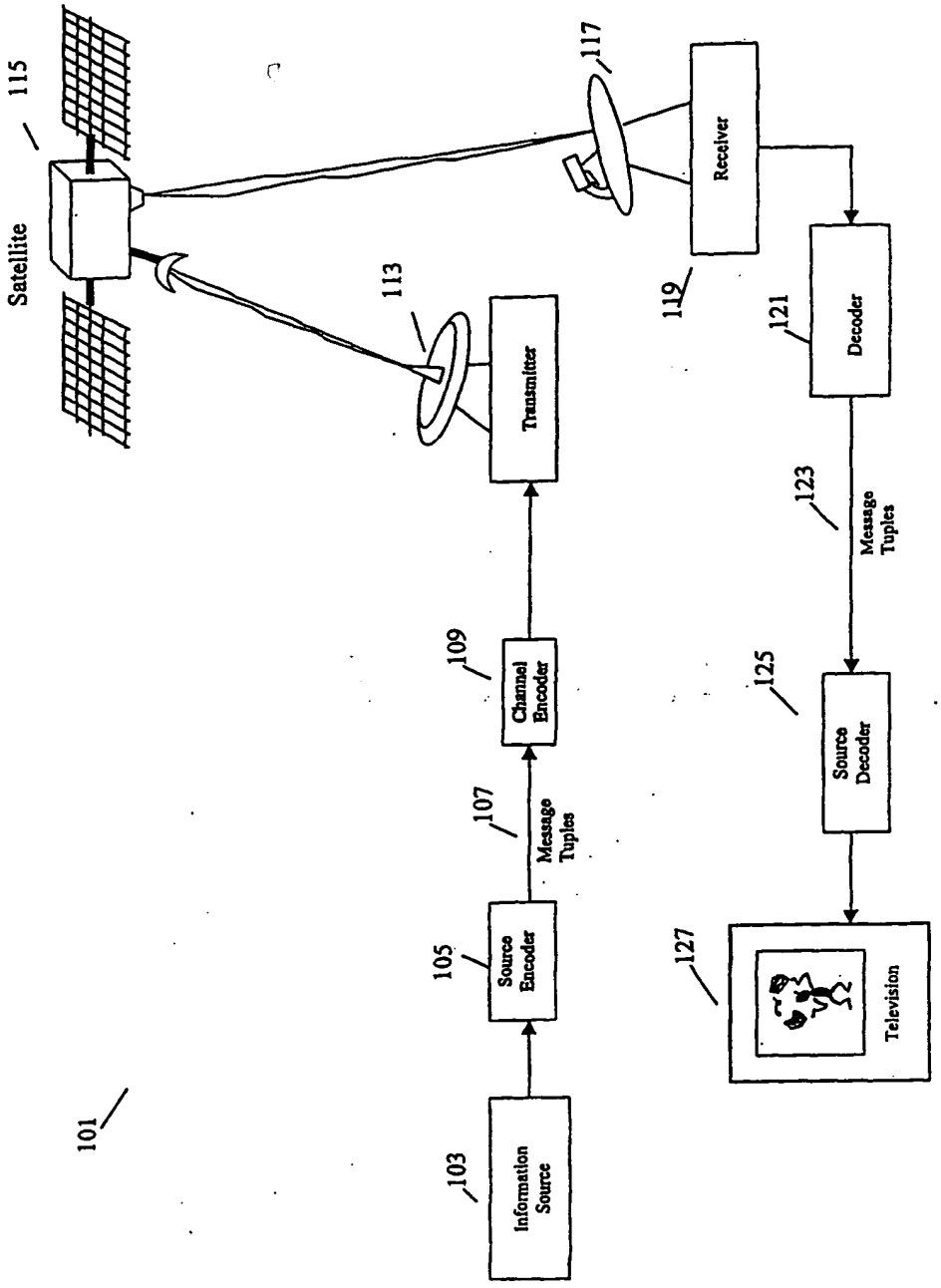

101
Information Source
103
Source Encoder
105
Message Tuples
107
Channel Encoder
109
Transmitter
113
Satellite
115
Receiver
117
119
Decoder
121
Message Tuples
123
Source Decoder
125
Television
127

Figure 1

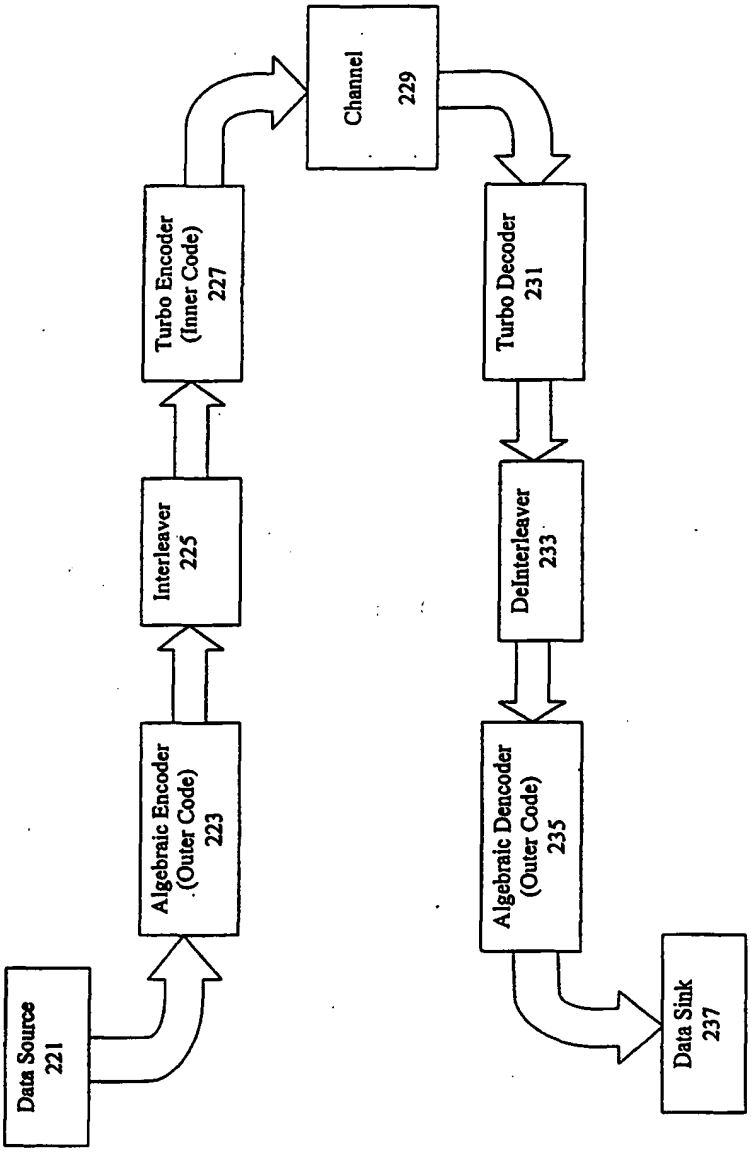

Data Source
221
Algebraic Encoder
(Outer Code)
223
Interleaver
225
Turbo Encoder
(Inner Code)
227
Channel
229
Turbo Decoder
231
DeInterleaver
233
Algebraic Dencoder
(Outer Code)
235
Data Sink
237

Figure 2A

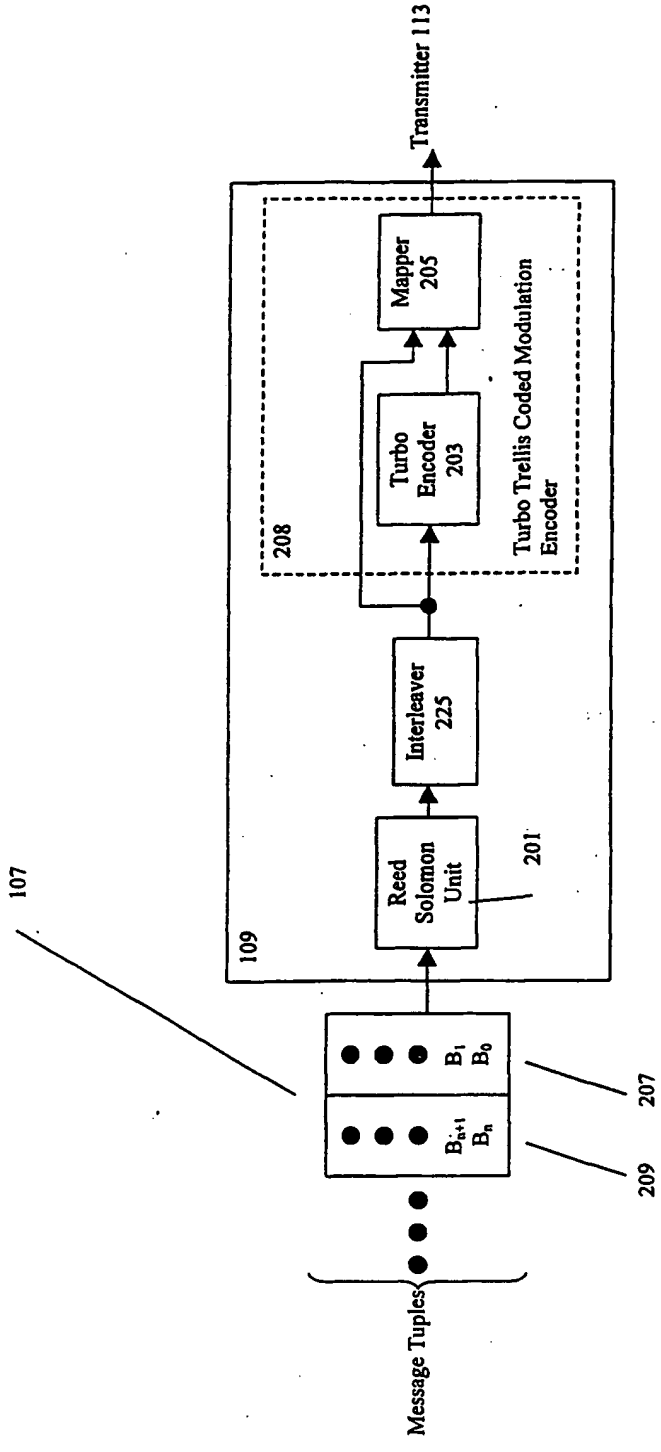

Transmitter 113
Mapper 205
Turbo Encoder 203
208
Turbo Trellis Coded Modulation Encoder
Interleaver 225
Reed Solomon Unit
201
109
107
B_1 B_0
B_{n+1} B_n
207
209
Message Tuples

Figure 2B

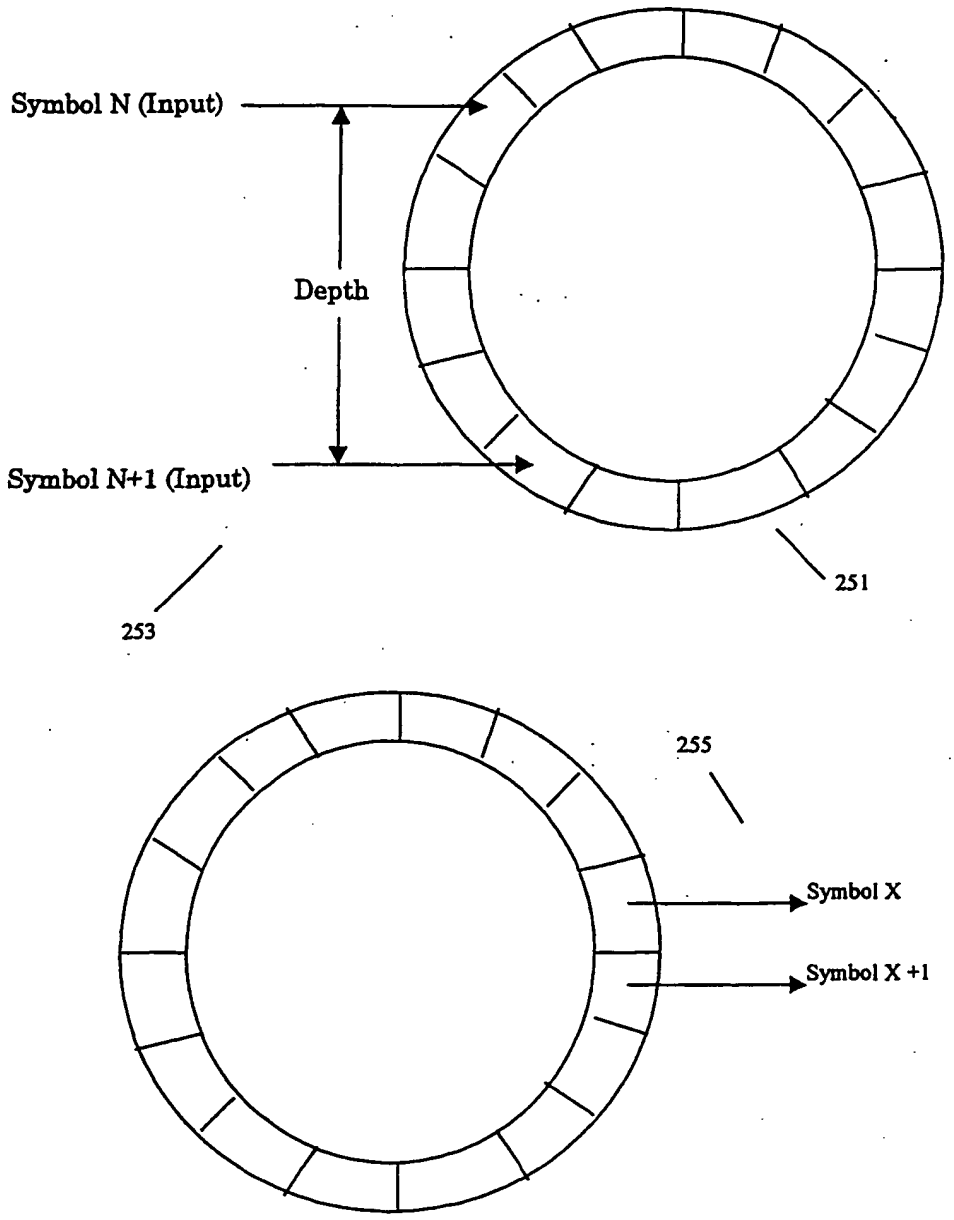

Symbol N (Input)
Depth
Symbol N+1 (Input)
253
251
255
Symbol X
Symbol X +1

Figure 2C

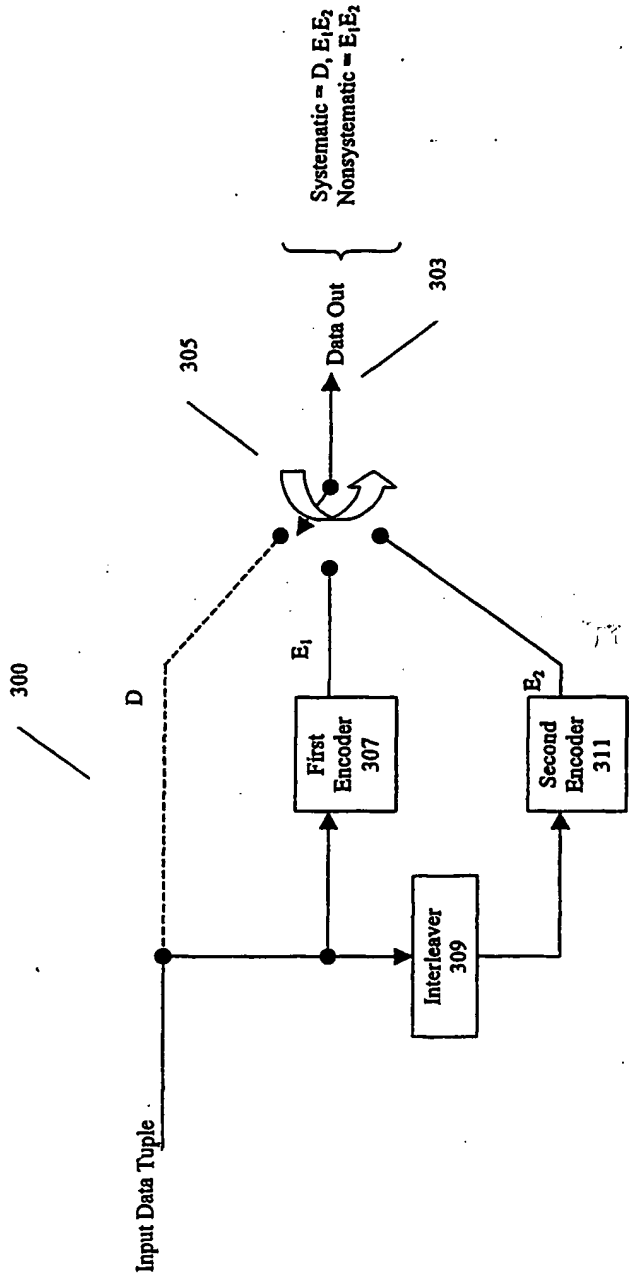

Systematic = D, E1E2
Nonsystematic = E1E2
Data Out
303
305
300
D
E1
E2
First Encoder 307
Second Encoder 311
Interleaver 309
Input Data Tuple

Figure 3

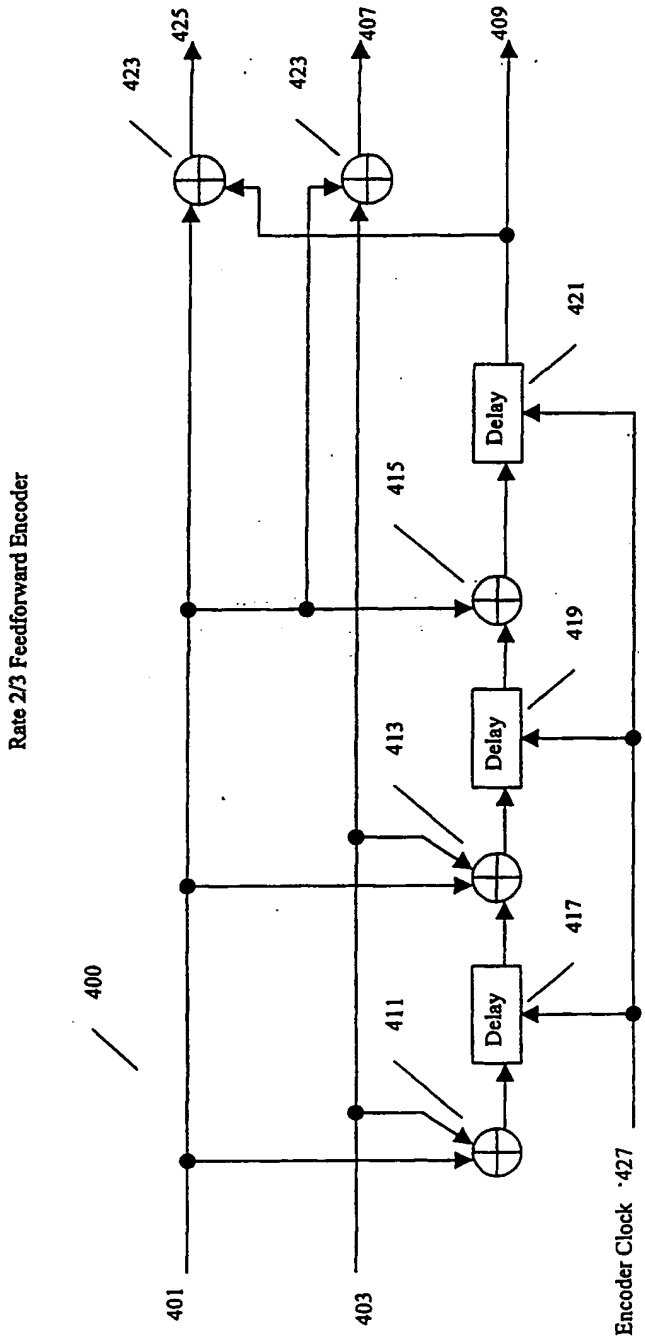

Rate 2/3 Feediforward Encoder
400
401
403
407
409
411
413
415
417
419
421
423
425
427
Delay
Delay
Delay
Encoder Clock

Figure 4

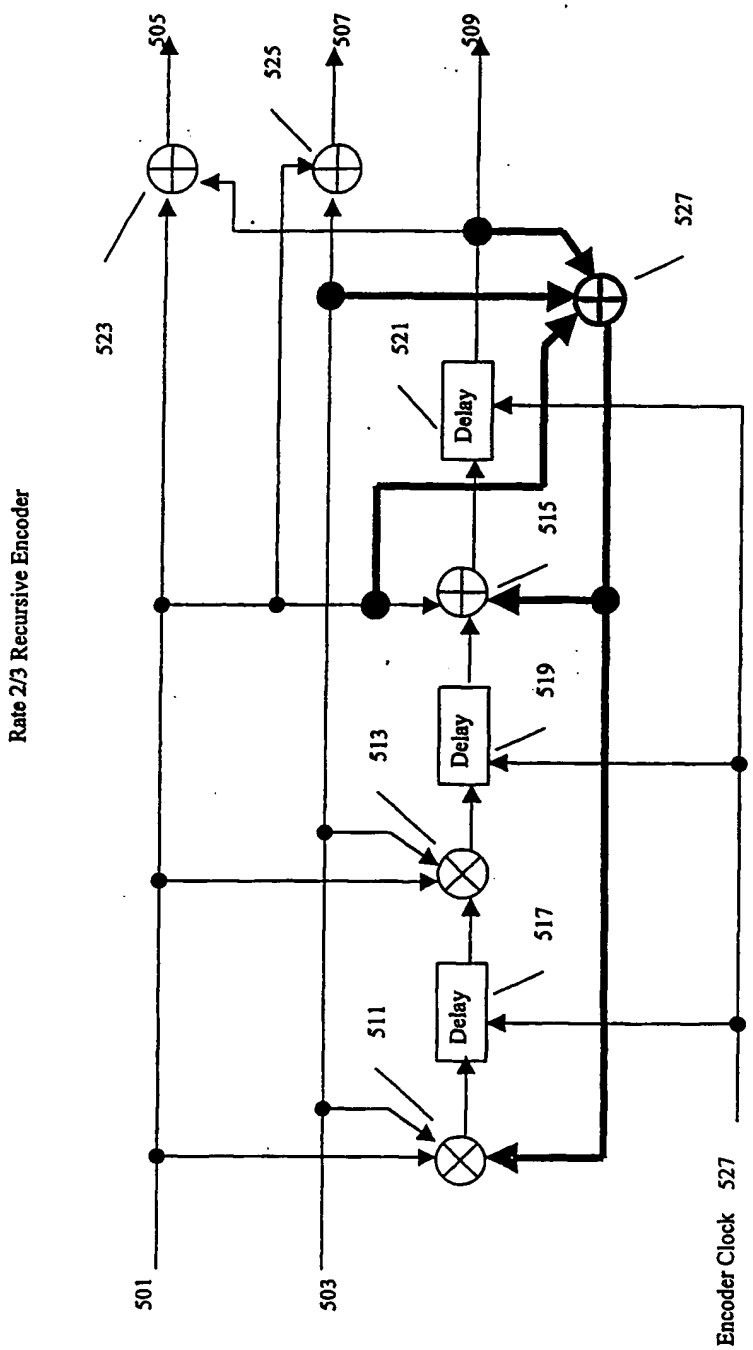
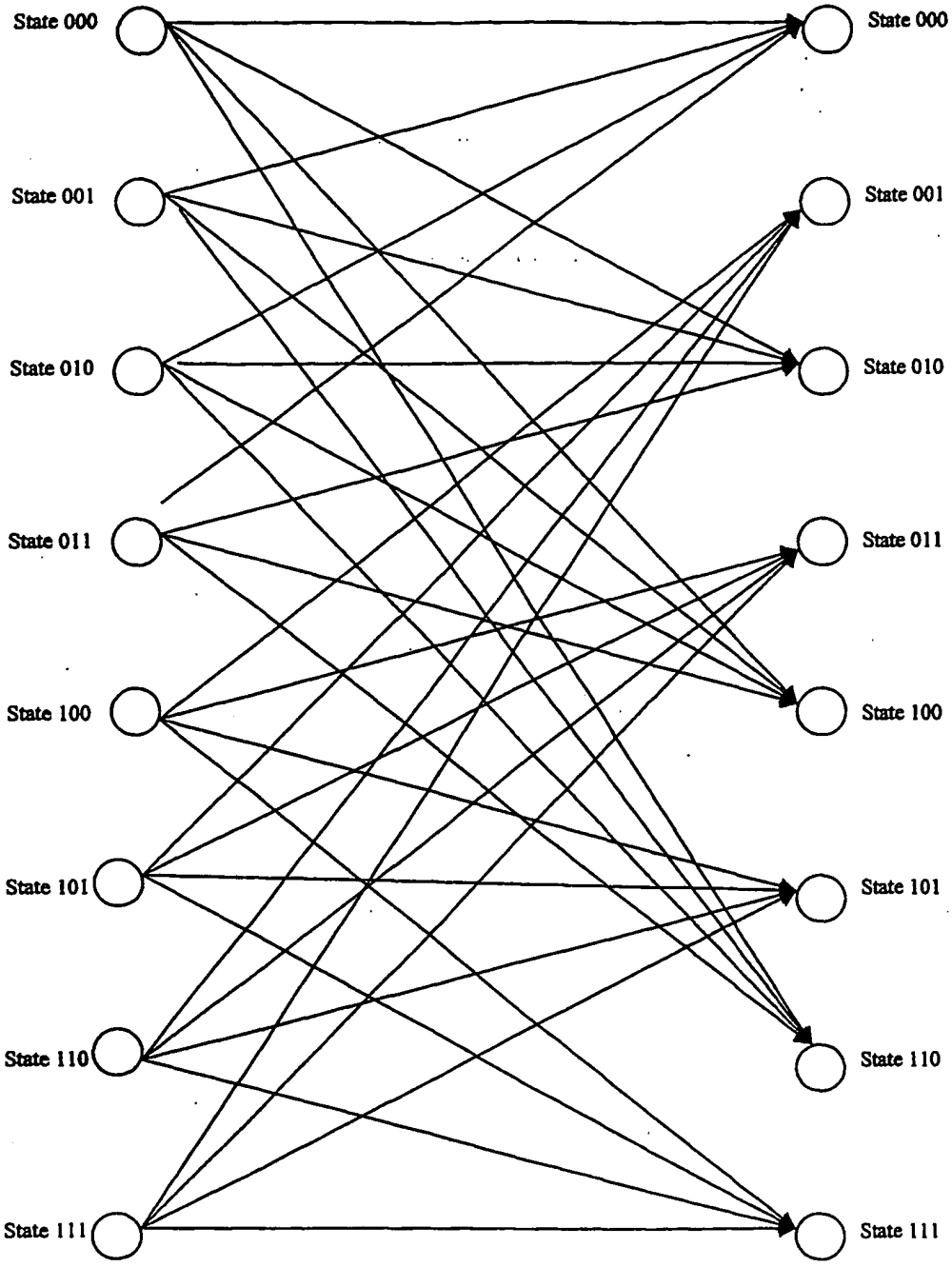

Rate 2/3 Recursive Encoder
505
525
507
509
523
527
521
Delay
515
519
Delay
513
517
Delay
511
501
503
Encoder Clock
527

Figure 5



Rate 2/3 basic constituent encoder

Figure 6

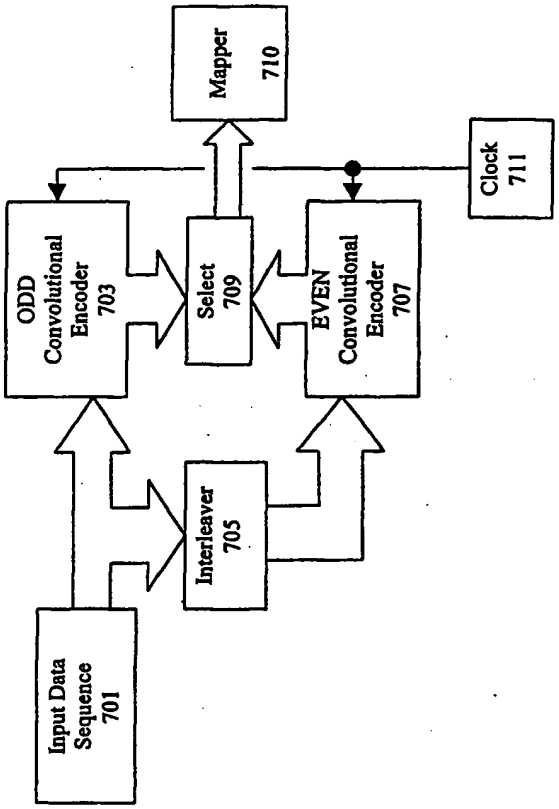

Mapper
710
Clock
711
ODD
Convolutional
Encoder
703
Select
709
EVEN
Convolutional
Encoder
707
Interleaver
705
Input Data
Sequence
701

Figure 7

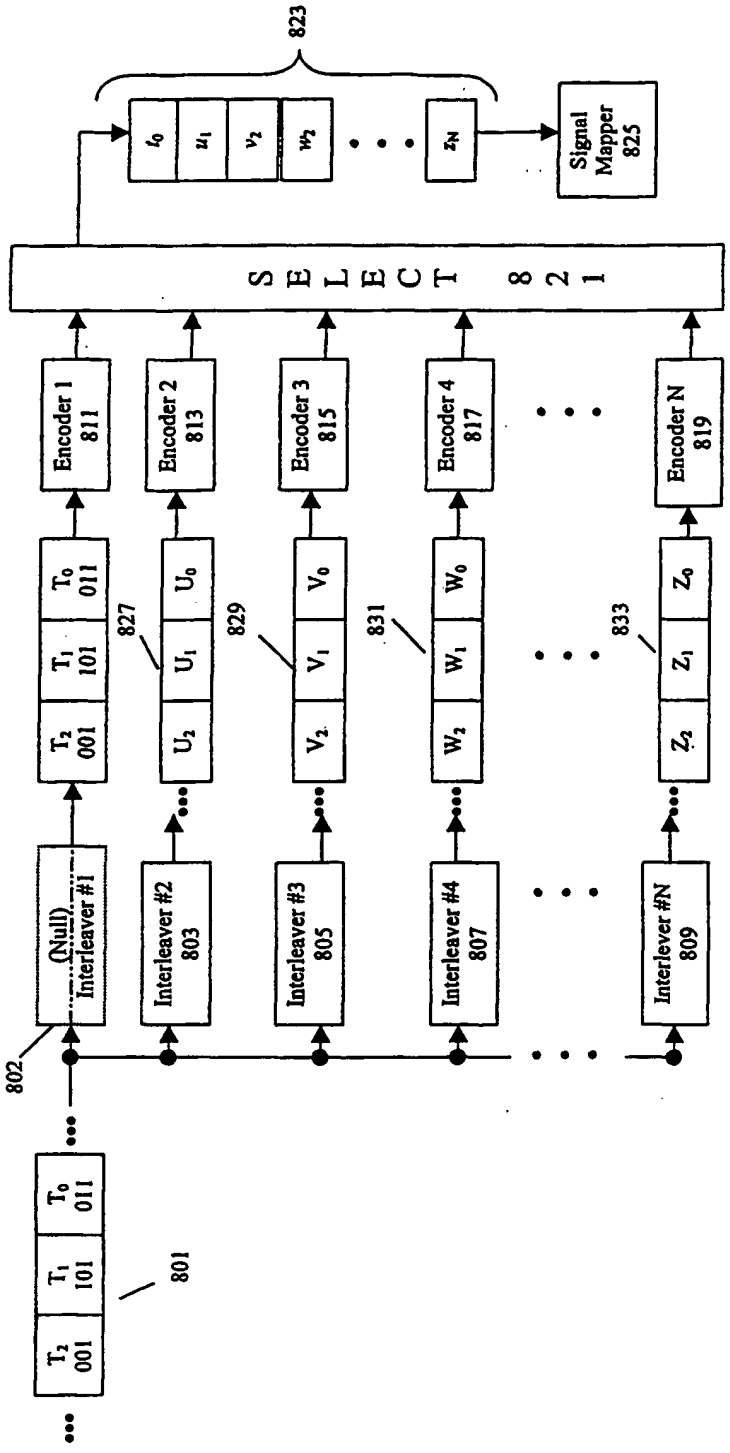

823
t_0 u_1 v_2 w_2 z_N
Signal Mapper 825
SELECT 821
Encoder 1 811
Encoder 2 813
Encoder 3 815
Encoder 4 817
Encoder N 819
T_0 011
T_1 101
T_2 001
U_0 U_1 U_2
V_0 V_1 V_2
W_0 W_1 W_2
Z_0 Z_1 Z_2
827
829
831
833
(Null) Interleaver #1
Interleaver #2 803
Interleaver #3 805
Interleaver #4 807
Interlever #N 809
802
801

Figure 8A

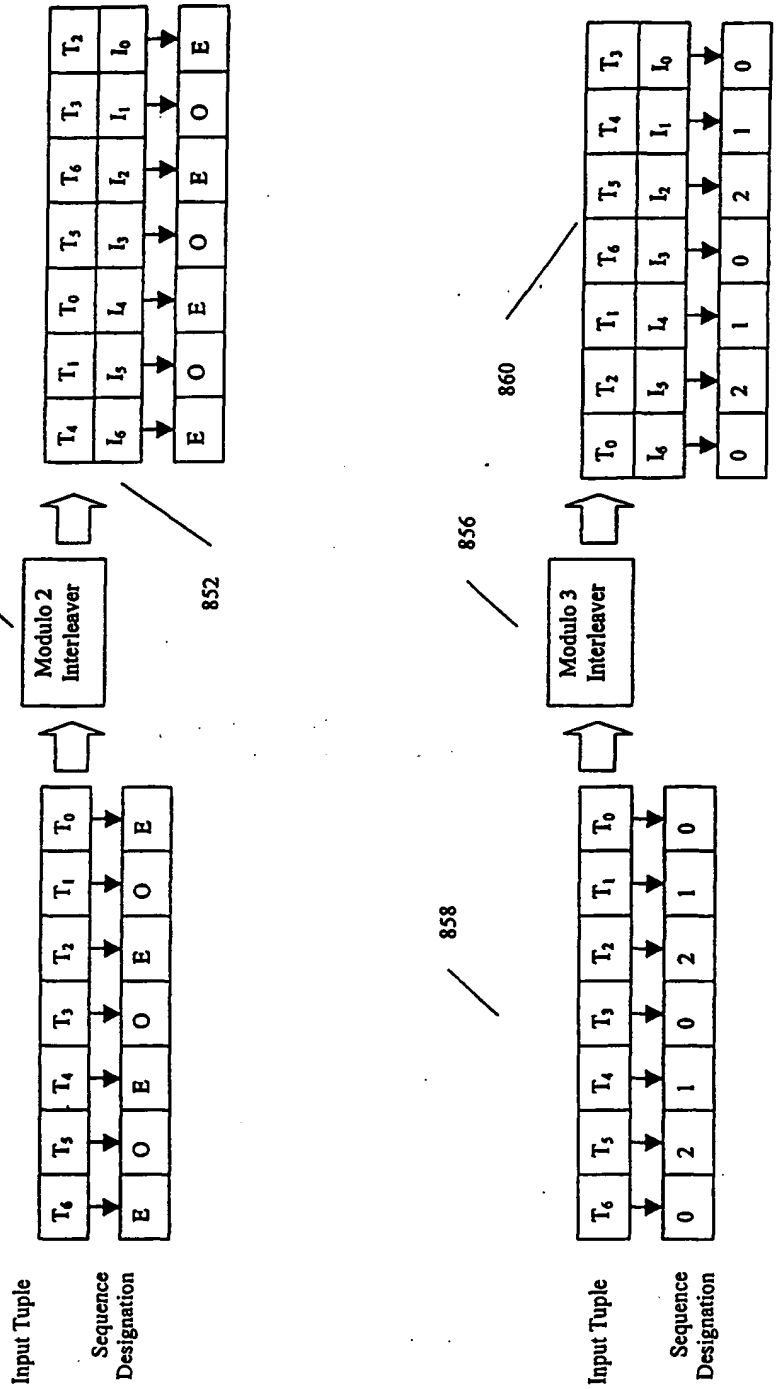

Input Tuple
Sequence Designation
Modulo 2 Interleaver
Modulo 3 Interleaver
852
854
856
858
860

Figure 8B

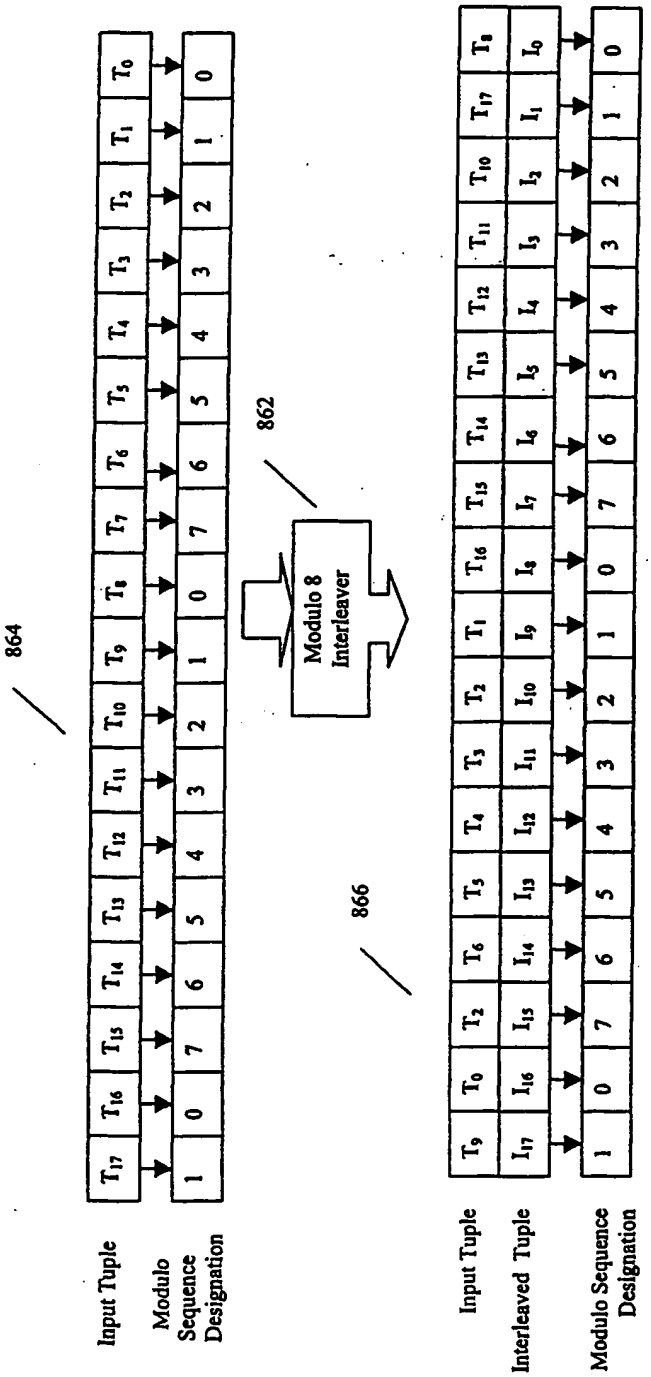

864
862
866
Modulo 8 Interleaver
Input Tuple
Modulo Sequence Designation
Input Tuple
Interleaved Tuple
Modulo Sequence Designation

Figure 8C

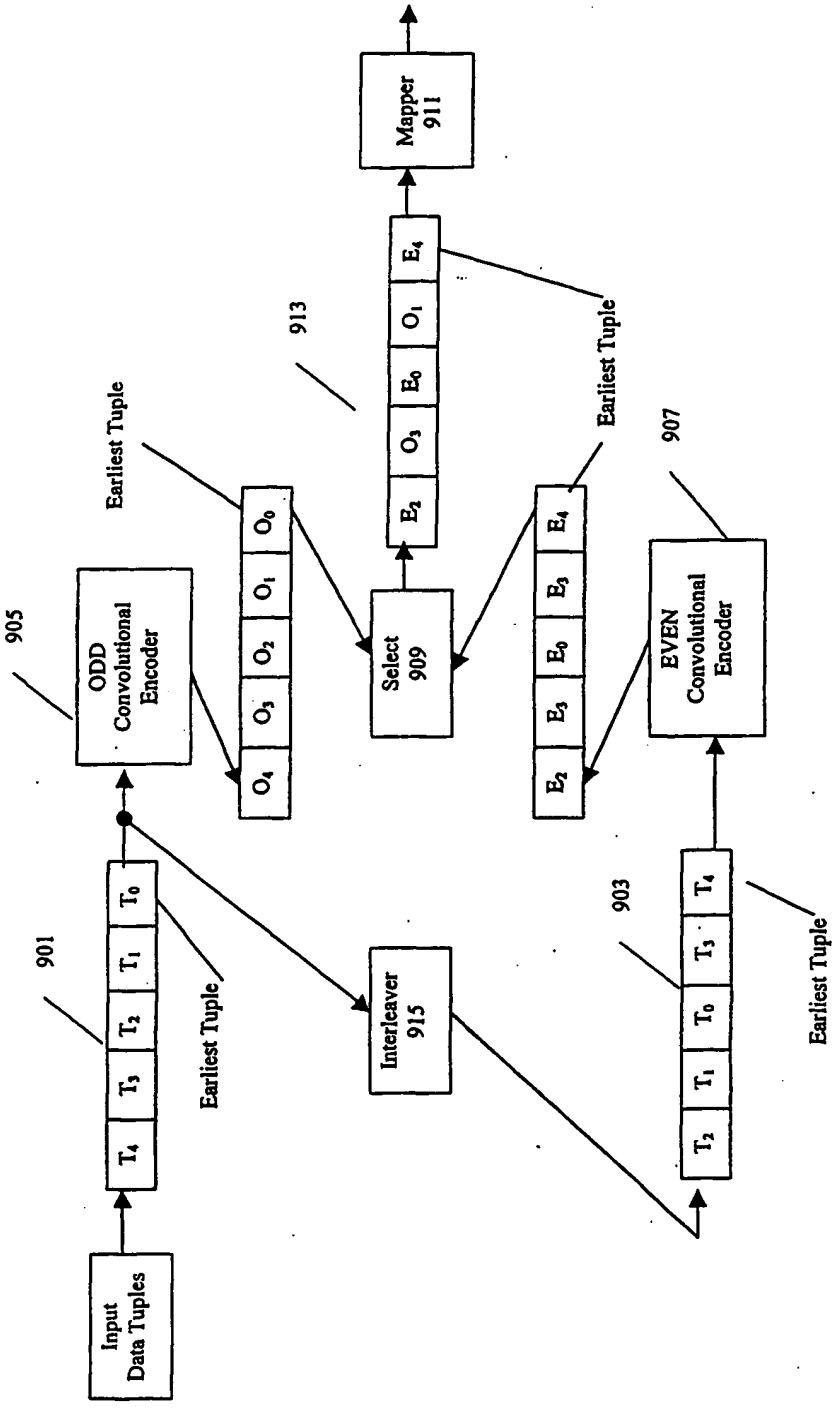

Input Data Tuples
901
T4 T3 T2 T1 T0
Earliest Tuple
905
ODD Convolutional Encoder
O4 O3 O2 O1 O0
Earliest Tuple
Interleaver 915
903
T2 T1 T0 T3 T4
Earliest Tuple
907
EVEN Convolutional Encoder
E2 E3 E0 E5 E4
Earliest Tuple
Select 909
913
E2 O3 E0 O1 E4
Mapper 911

Figure 9

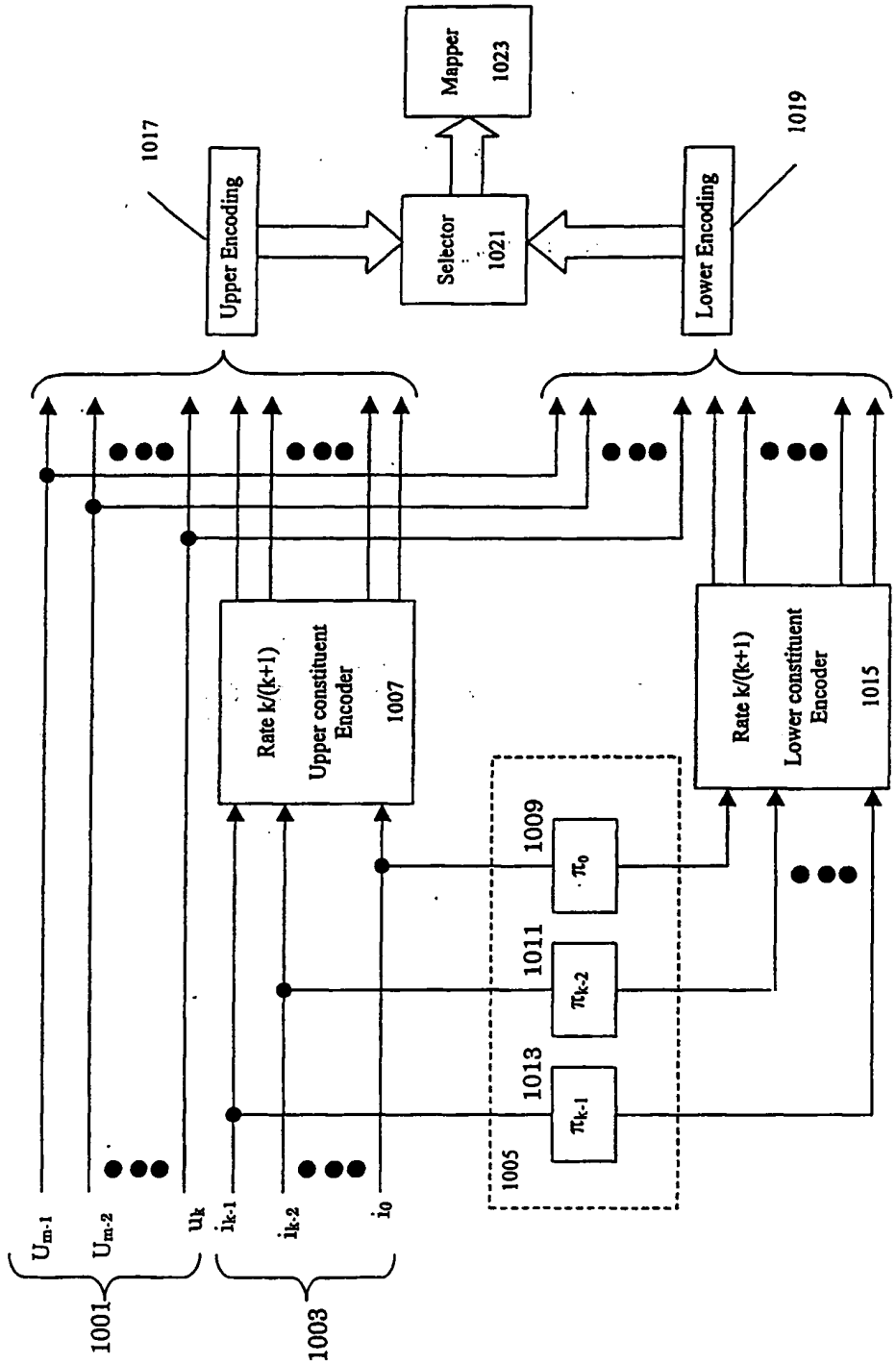

Mapper 1023
1017
Upper Encoding
Selector 1021
Lower Encoding
1019
Rate k/(k+1) Upper constituent Encoder 1007
Rate k/(k+1) Lower constituent Encoder 1015
1009
π_0
1011
π_{k-2}
1013
π_{k-1}
1005
U_{m-1}
U_{m-2}
u_k
i_{k-1}
i_{k-2}
i_0
1001
1003

Figure 10

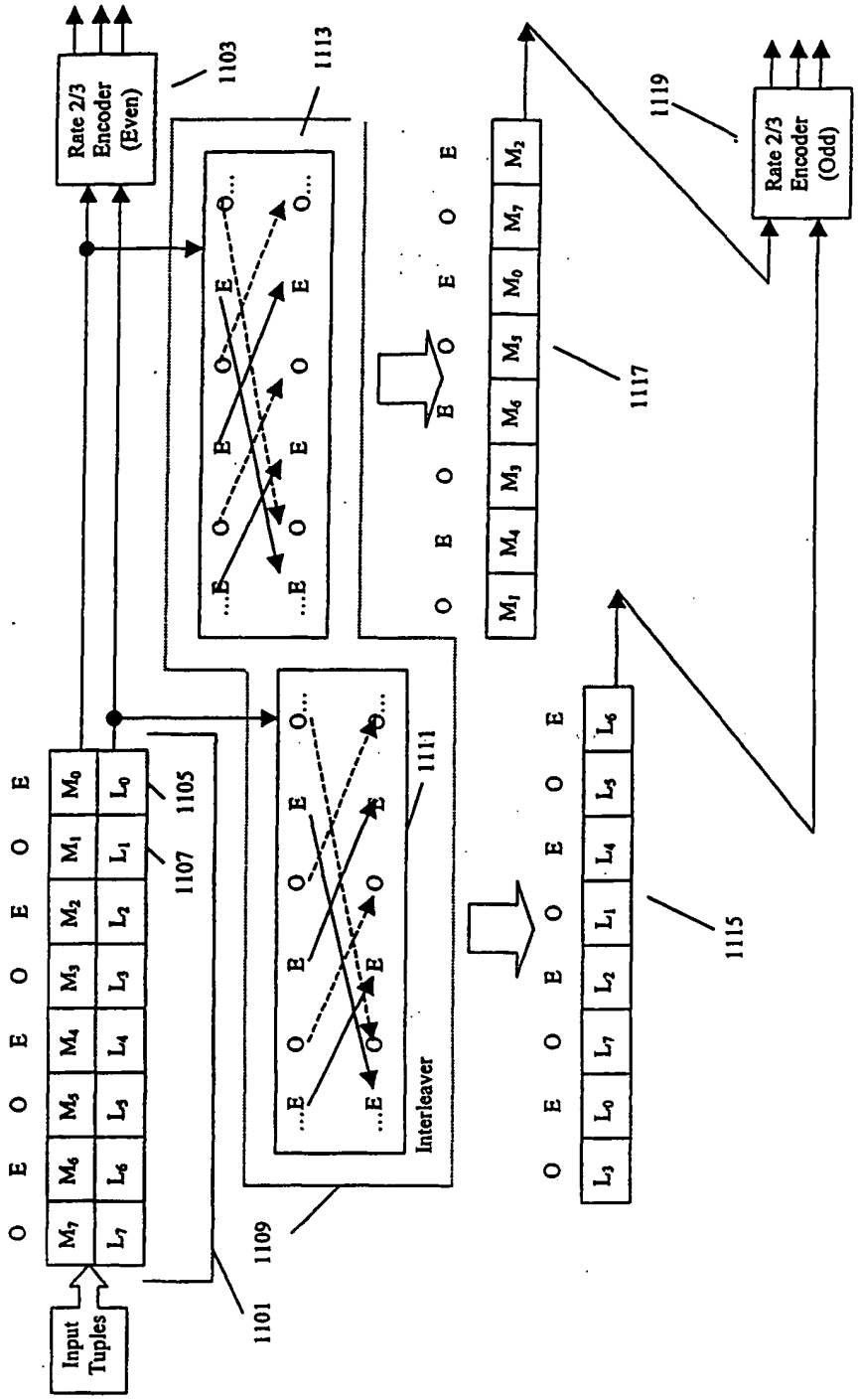

Rate 2/3 Encoder (Even)
1103
1113
Rate 2/3 Encoder (Odd)
1119
1117
1111
Interleaver
1105
1107
1109
1101
1115
Input Tuples
O E O E O E O E
O E O E O E O E
O E O E O E O E
O E O E O E O E

Figure 11A

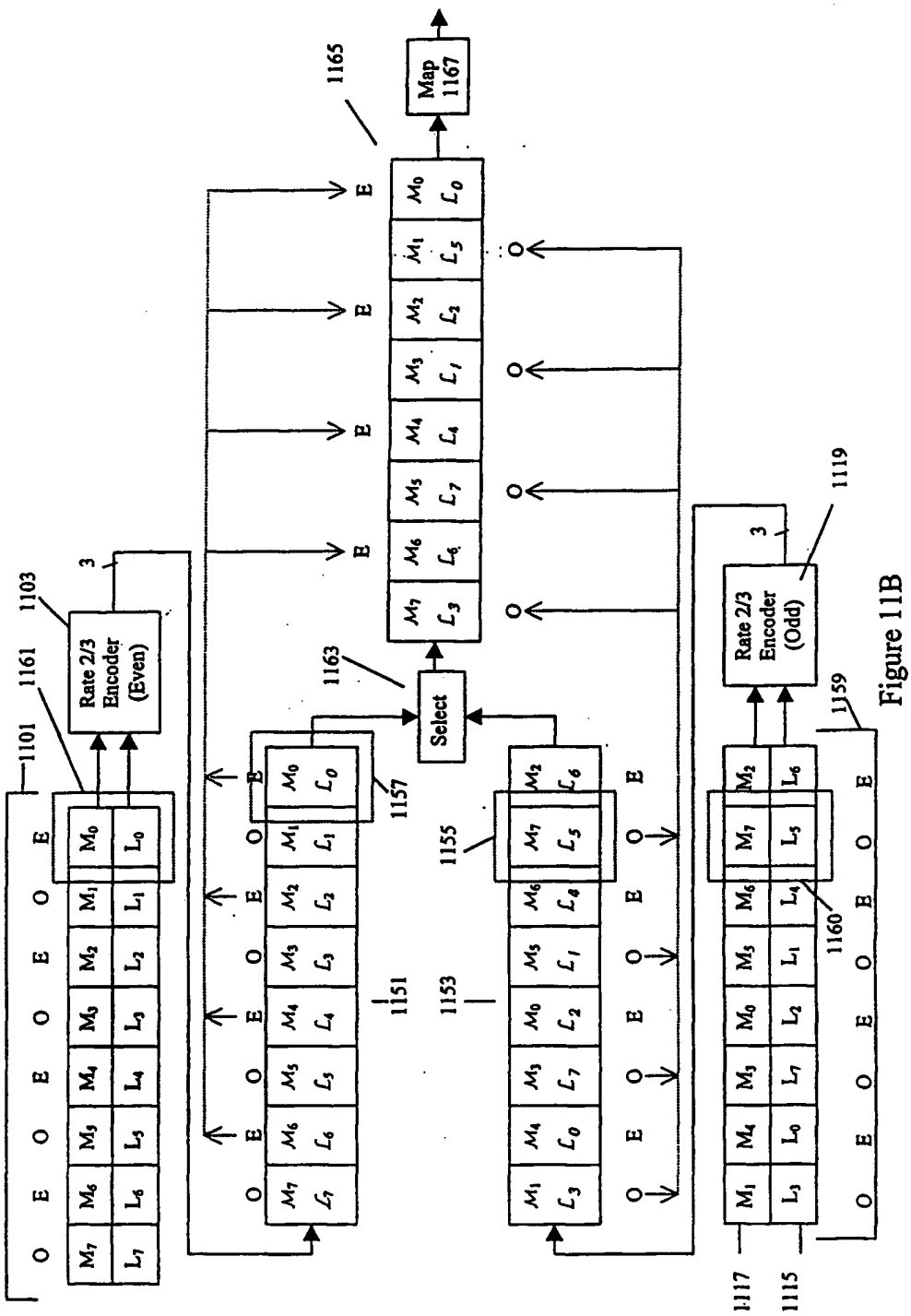

1165
Map 1167
1163
Select
1103
Rate 2/3 Encoder (Even)
1161
1101
1157
1151
1153
1155
1119
Rate 2/3 Encoder (Odd)
1159
1160
1117
1115
3
E
O

Figure 11B

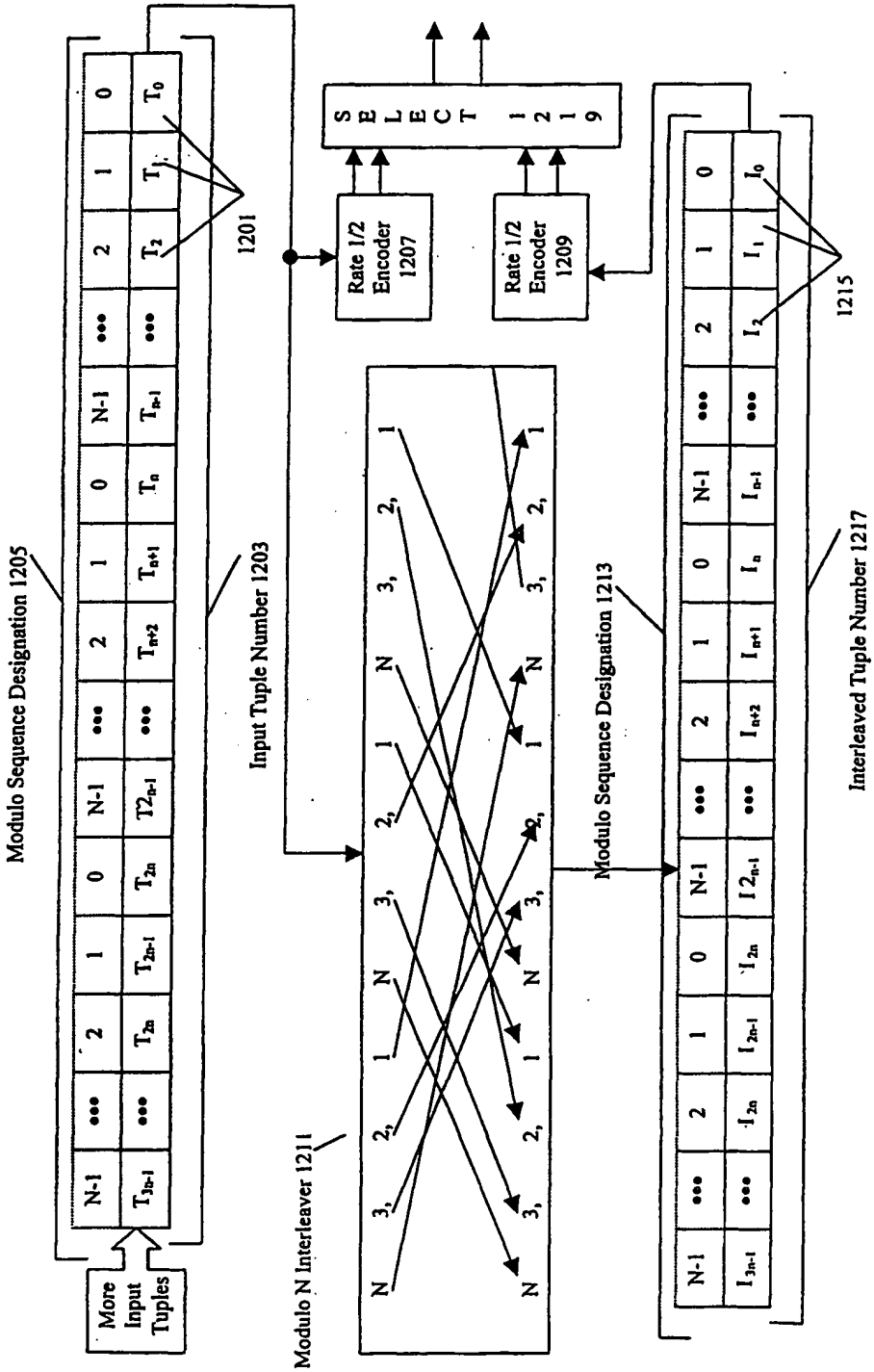

Modulo Sequence Designation 1205
Input Tuple Number 1203
1201
Rate 1/2 Encoder 1207
Rate 1/2 Encoder 1209
SELECT 1219
Modulo N Interleaver 1211
Modulo Sequence Designation 1213
Interleaved Tuple Number 1217
1215
More Input Tuples

Figure 12

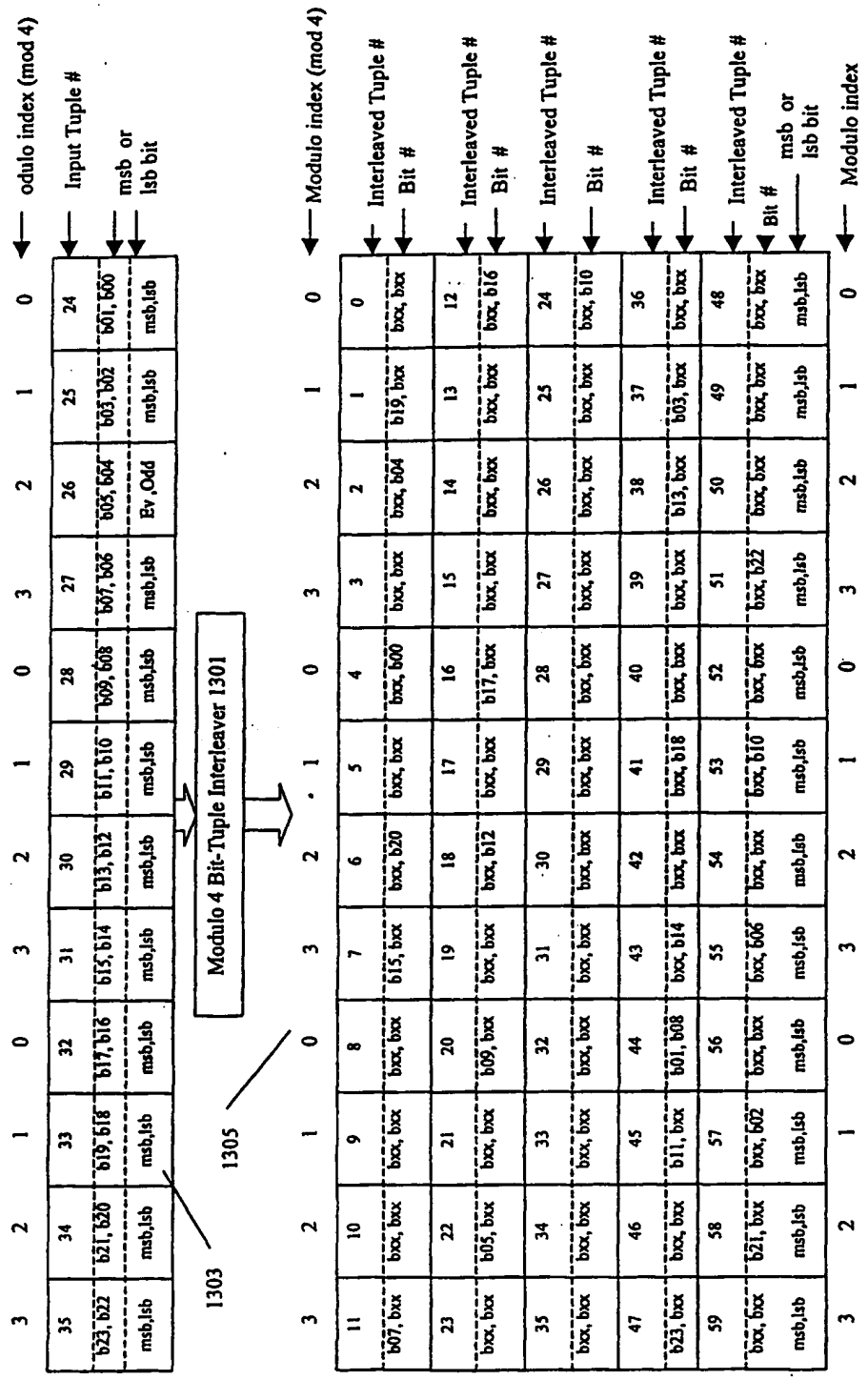

odulo index (mod 4)
Input Tuple #
msb or lsb bit
Modulo 4 Bit-Tuple Interleaver 1301
1303
1305
Modulo index (mod 4)
Interleaved Tuple #
Bit #
Modulo index

Figure # 13

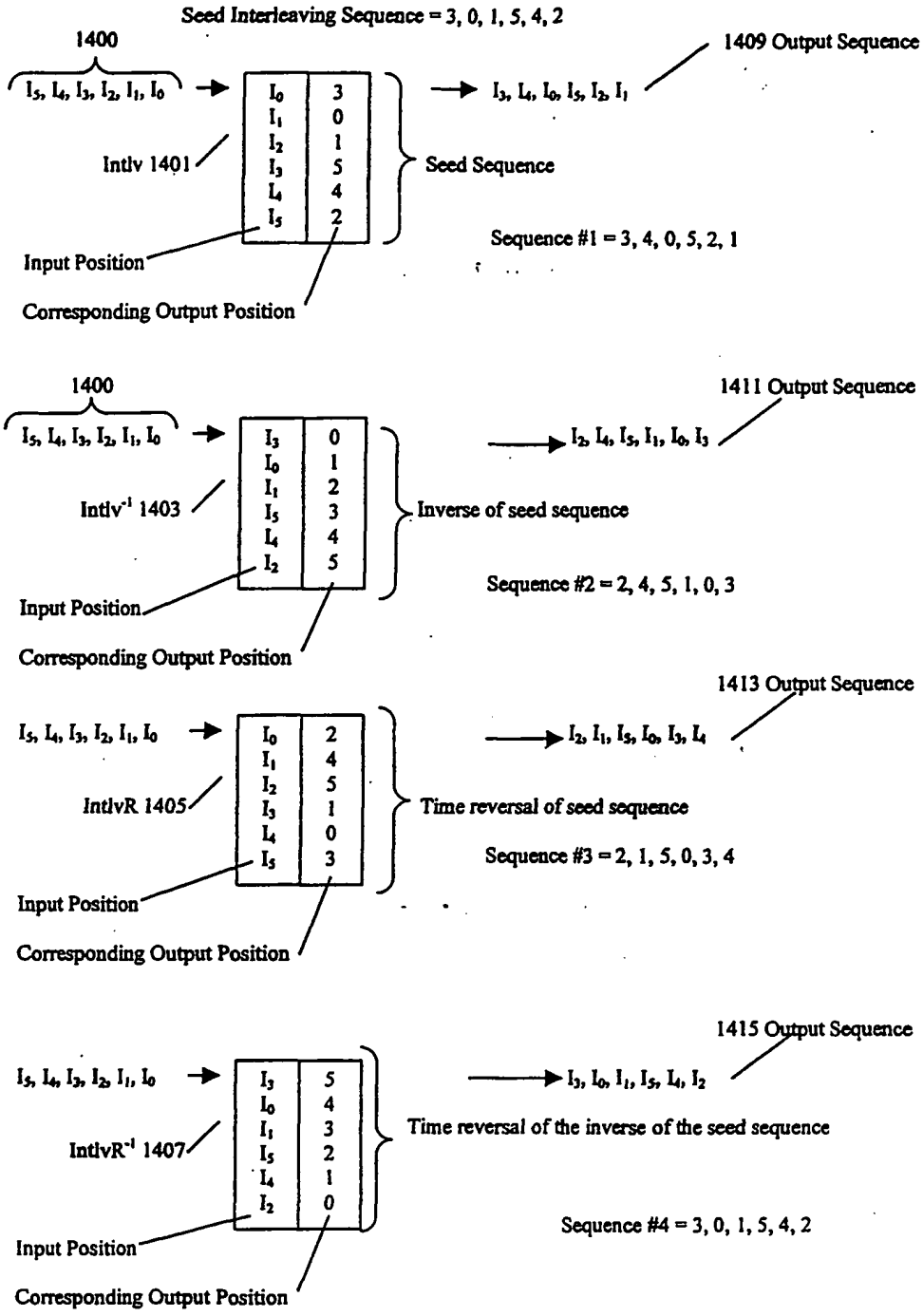

Seed Interleaving Sequence = 3, 0, 1, 5, 4, 2
1400
$I_5, I_4, I_3, I_2, I_1, I_0$
Intlv 1401
Input Position
Corresponding Output Position
I_0 3
I_1 0
I_2 1
I_3 5
I_4 4
I_5 2
Seed Sequence
$I_3, I_4, I_0, I_5, I_2, I_1$
1409 Output Sequence
Sequence #1 = 3, 4, 0, 5, 2, 1
1400
$I_5, I_4, I_3, I_2, I_1, I_0$
$Intlv^{-1}$ 1403
Input Position
Corresponding Output Position
I_3 0
I_0 1
I_1 2
I_5 3
I_4 4
I_2 5
Inverse of seed sequence
$I_2, I_4, I_5, I_1, I_0, I_3$
1411 Output Sequence
Sequence #2 = 2, 4, 5, 1, 0, 3
$I_5, I_4, I_3, I_2, I_1, I_0$
IntlvR 1405
Input Position
Corresponding Output Position
I_0 2
I_1 4
I_2 5
I_3 1
I_4 0
I_5 3
Time reversal of seed sequence
$I_2, I_1, I_5, I_0, I_3, I_4$
1413 Output Sequence
Sequence #3 = 2, 1, 5, 0, 3, 4
$I_5, I_4, I_3, I_2, I_1, I_0$
$IntlvR^{-1}$ 1407
Input Position
Corresponding Output Position
I_3 5
I_0 4
I_1 3
I_5 2
I_4 1
I_2 0
Time reversal of the inverse of the seed sequence
$I_3, I_0, I_1, I_5, I_4, I_2$
1415 Output Sequence
Sequence #4 = 3, 0, 1, 5, 4, 2

Figure 14A

Table 1

Seq. #1		3		4		0		5		2		1
Seq. #2	2		4		5		1		0		3	
Seq. 1-2	2	3	4	4	5	0	1	5	0	2	3	1

Table 2

Position	11	10	9	8	7	6	5	4	3	2	1	0
Seq. 1-2	2	3	4	4	5	0	1	5	0	2	3	1
X2 (Mod)	4	6	8	8	10	0	2	10	0	4	6	2
+ mod 2 (Position)	+1	+0	+1	+0	+1	+0	+1	+0	+1	+0	+1	+0
Seq. 5	5	6	9	8	11	0	3	10	1	4	7	2

Table 3

Seq. #3		2		1		5		0		3		4
Seq. #4	3		0		1		5		4		2	
Seq. 3-4	3	2	0	1	1	5	5	0	4	3	2	4

Table 4

Position	11	10	9	8	7	6	5	4	3	2	1	0
Seq.3-4	3	2	0	1	1	5	5	0	4	3	2	4
X2 (Mod)	6	4	0	2	2	10	10	0	8	6	4	8
+ mod 2 (Position)	+1	+0	+1	+0	+1	+0	+1	+0	+1	+0	+1	+0
Seq. 6	7	4	1	2	3	10	11	0	9	6	5	8

Table 5

Seq. #1			3			4			0			5			2			1
Seq. #2		2			4			5			1			0			3	
Seq. #3	2			1			5			0			3			4		
Seq.1-2-3	2	2	3	1	4	4	5	5	0	0	1	5	3	0	2	4	3	1

Table 6

Position	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Seq 1-2-3	2	2	3	1	4	4	5	5	0	0	1	5	3	0	2	4	3	1
X3 (Mod)	6	6	9	3	12	12	15	15	0	0	3	15	9	0	6	12	9	3
+ mod 3 (Position)	2	1	0	2	1	0	2	1	0	2	1	0	2	1	0	2	1	0
Seq. 7	8	7	9	5	13	12	17	16	0	2	4	15	11	1	6	14	10	3

Figure 14B

Table 7

Seq. #1				3				4				0				5				2				1
Seq. #2			2				4				5				1				0				3	
Seq. #3		2				1				5				0				3				4		
Seq. #4	3				0				1				5				4				2			
Seq.1-2-3-4	3	2	2	3	0	1	4	4	1	5	5	0	5	0	1	5	4	3	0	2	2	4	3	1

Table 8

Position	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Seq 1-2-3-4	3	2	2	3	0	1	4	4	1	5	5	0	5	0	1	5	4	3	0	2	2	4	3	1
X4 (Mod)	12	8	8	12	0	4	16	16	4	20	20	0	20	0	4	20	16	12	0	8	8	16	12	4
+ mod 4 (Position)	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0
Seq. 8	15	10	9	12	3	6	17	16	7	22	21	0	23	2	5	20	19	14	1	8	11	18	13	4

Figure 14C

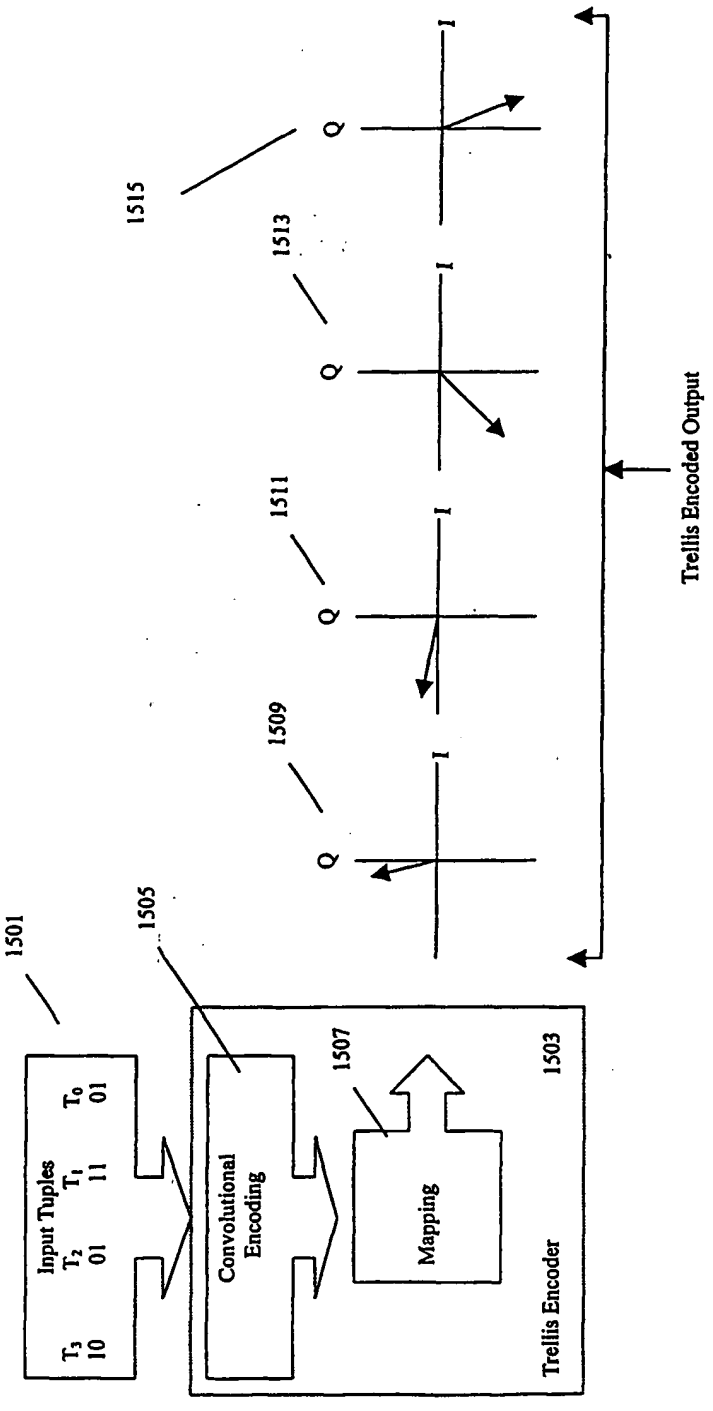

1501
1505
1507
1503
Input Tuples
T_0 01
T_1 11
T_2 01
T_3 10
Convolutional Encoding
Mapping
Trellis Encoder
1509
1511
1513
1515
I
Q
Trellis Encoded Output

Figure 15

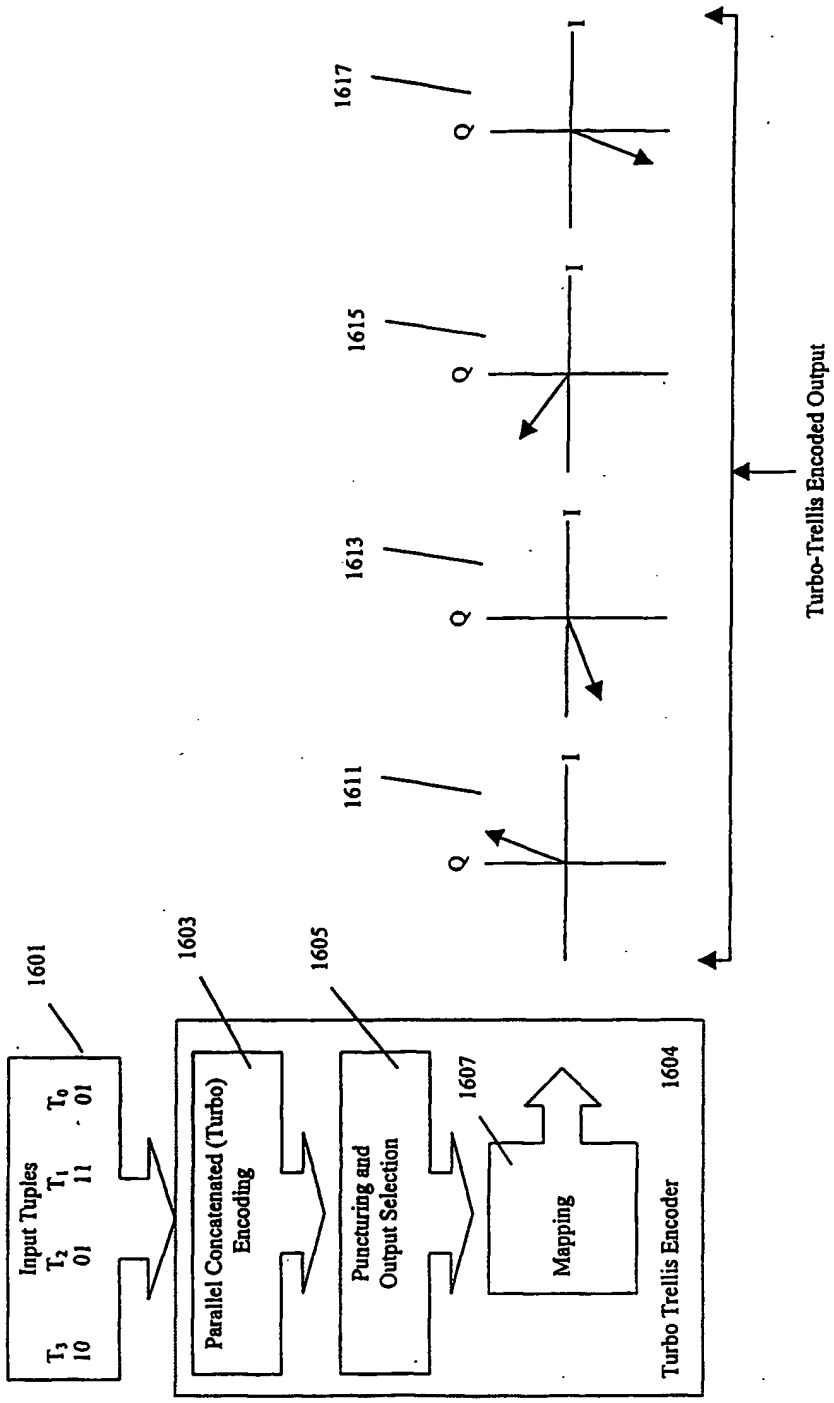

1601
Input Tuples
T3 10
T2 01
T1 11
T0 01
1603
Parallel Concatenated (Turbo) Encoding
1605
Puncturing and Output Selection
1607
Mapping
1604
Turbo Trellis Encoder
1611
1613
1615
1617
I
Q
Turbo-Trellis Encoded Output

Figure 16

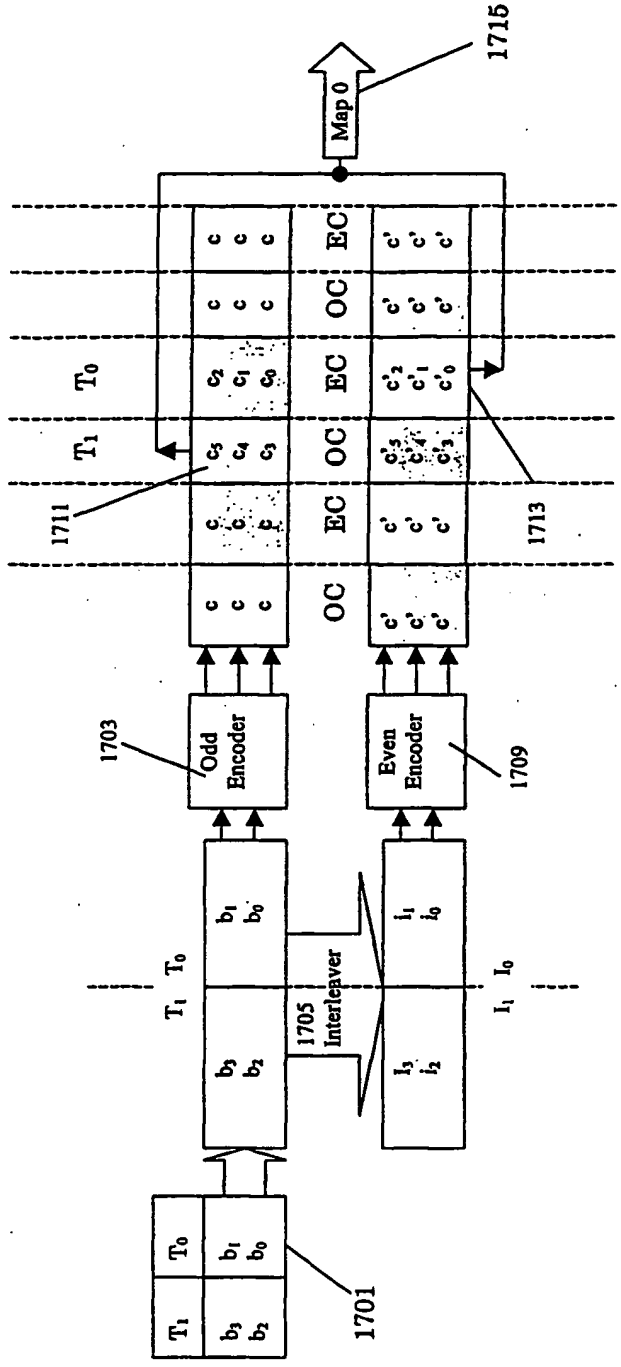


Figure 17

Rate 2-3 Encoder

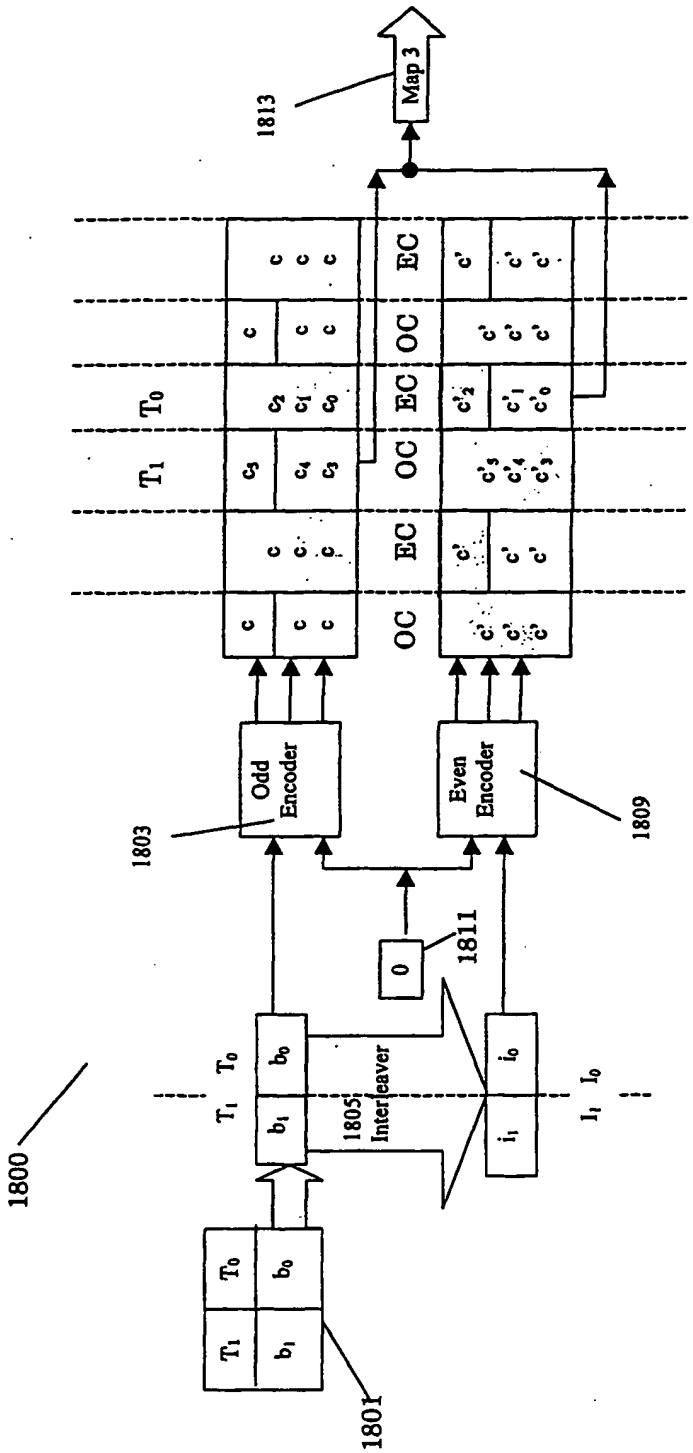


Figure 18A

Rate 1/2 Encoder

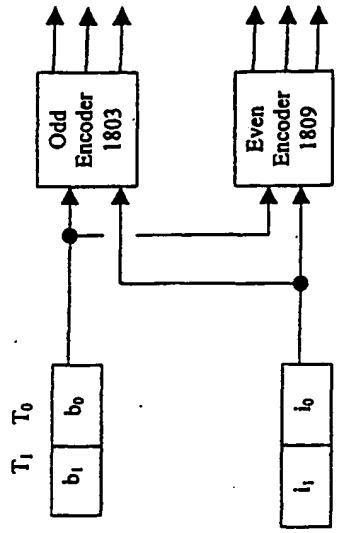


Figure 18 C

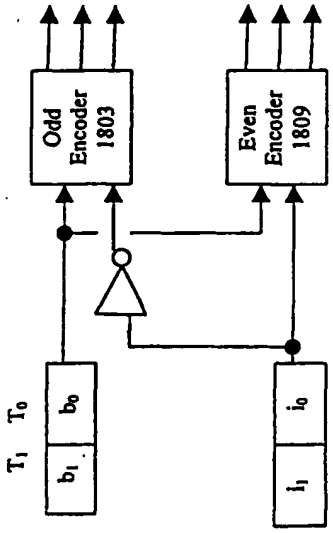


Figure 18 E

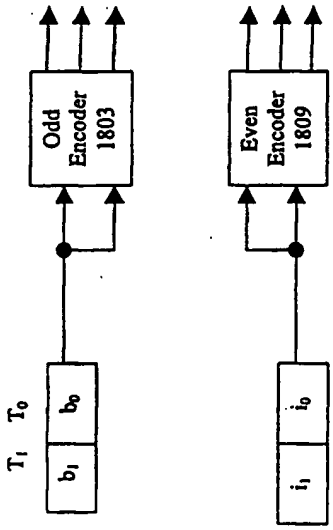


Figure 18 B

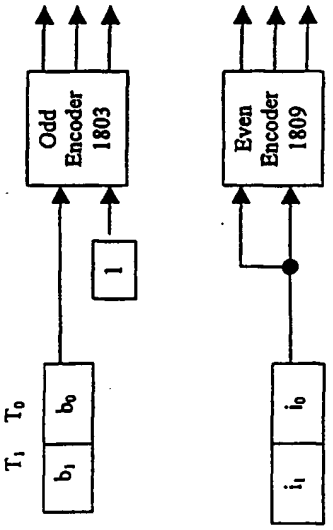
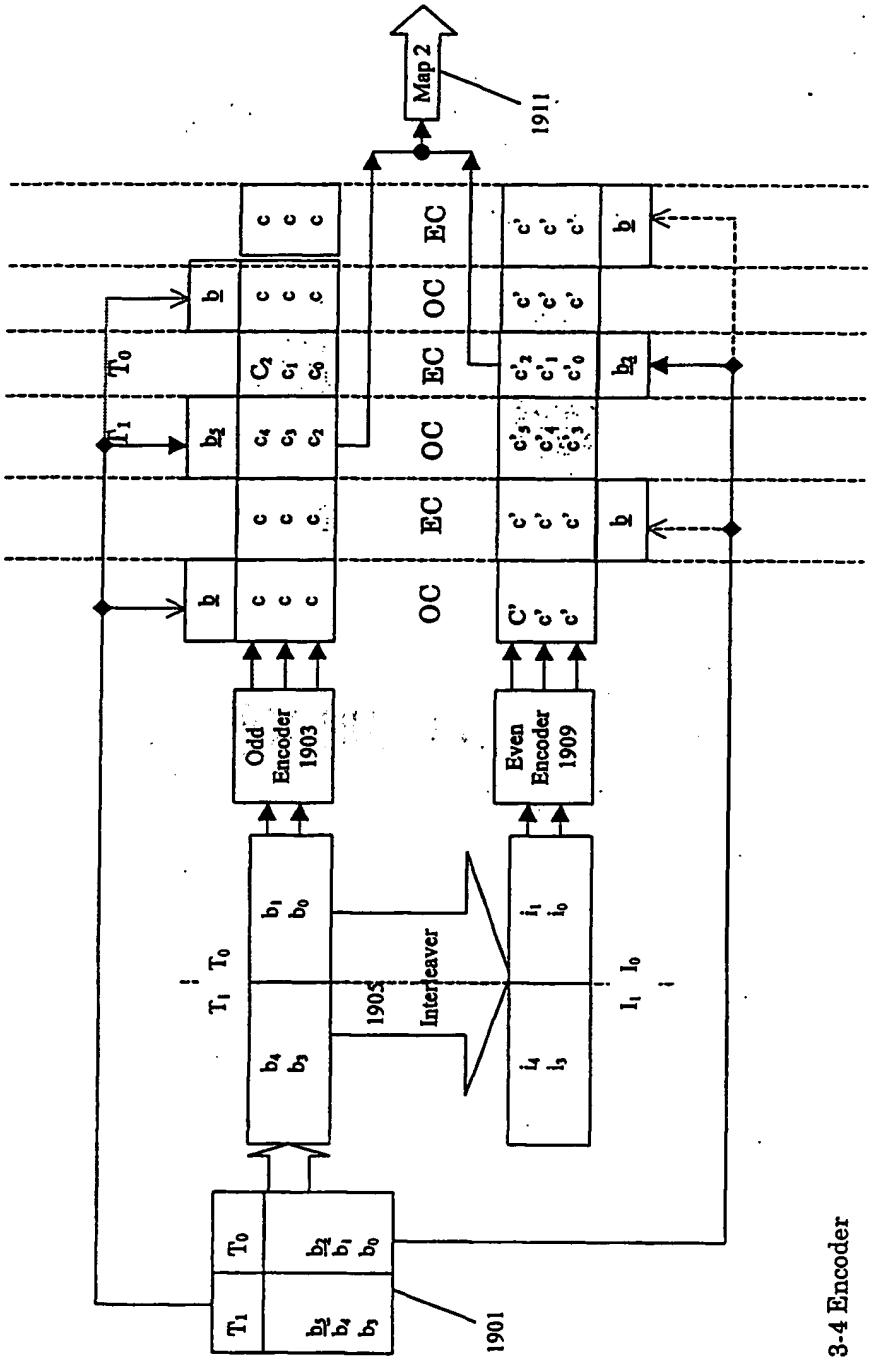


Figure 18 D



Rate 3-4 Encoder

Figure 19

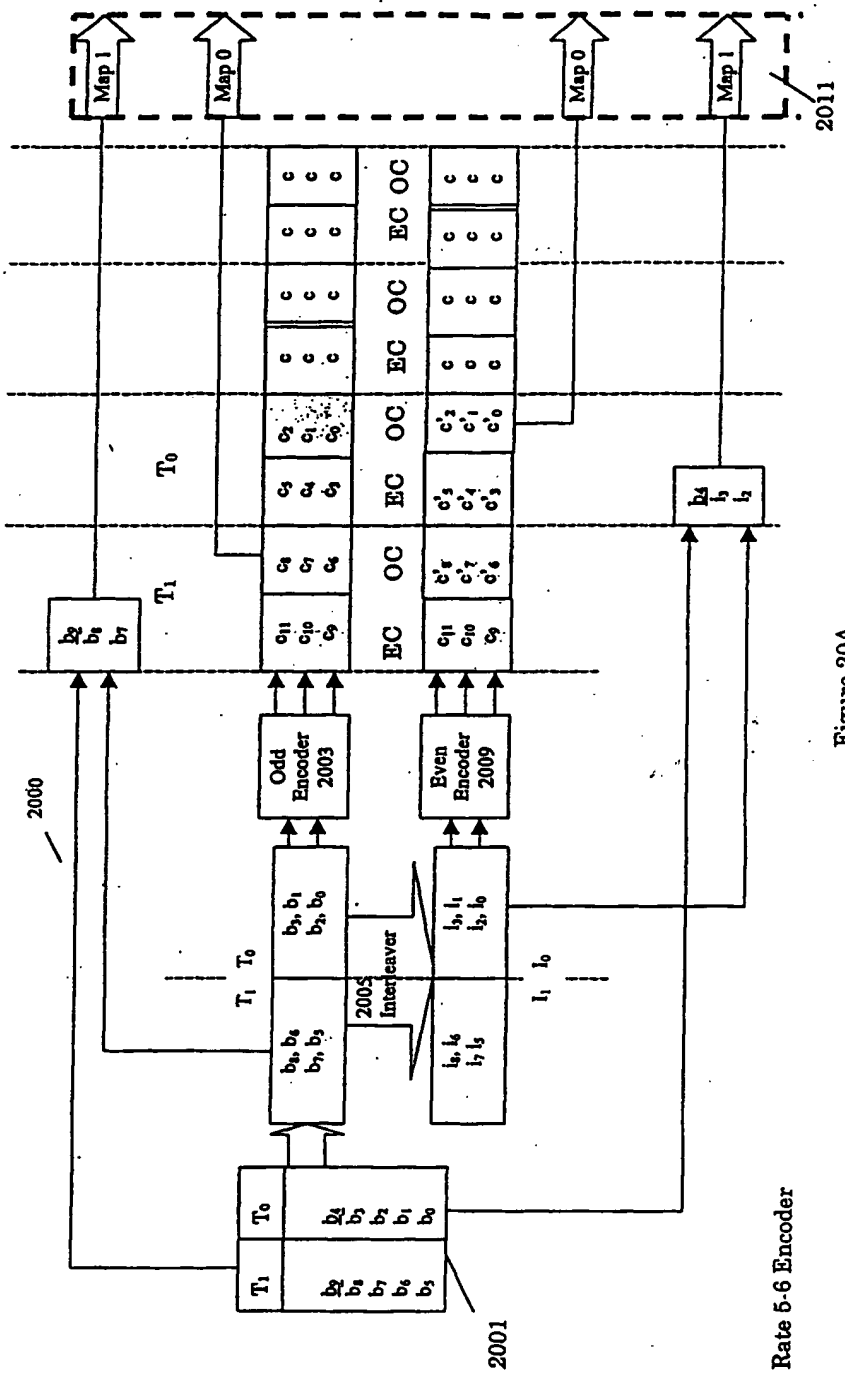

Map 1
Map 0
Map 0
Map 1
2011
EC
OC
T_0
T_1
Odd Encoder 2003
Even Encoder 2009
2000
2005 Interleaver
2001
Rate 5-6 Encoder

Figure 20A

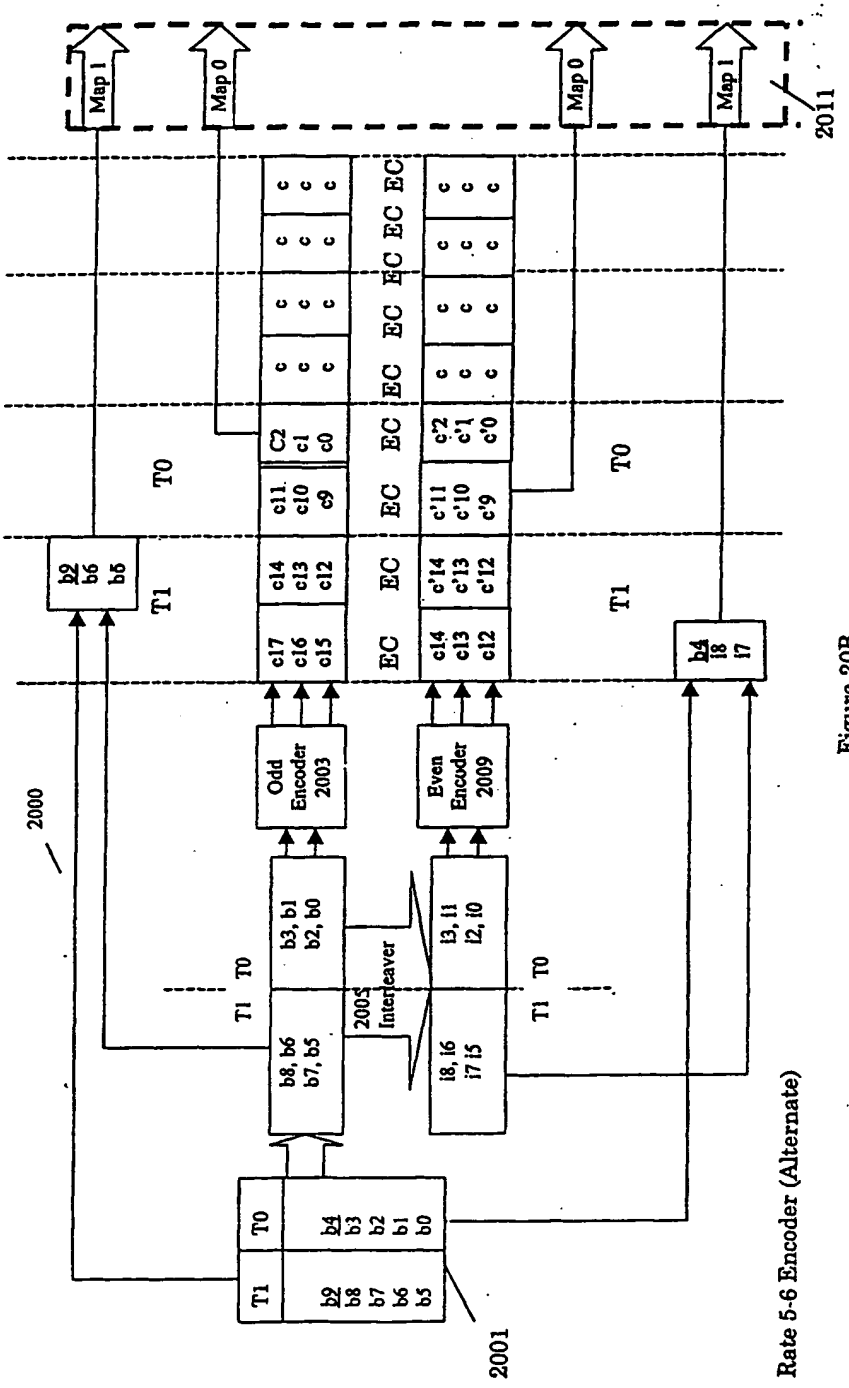


Figure 20B

Rate 5-6 Encoder (Alternate)

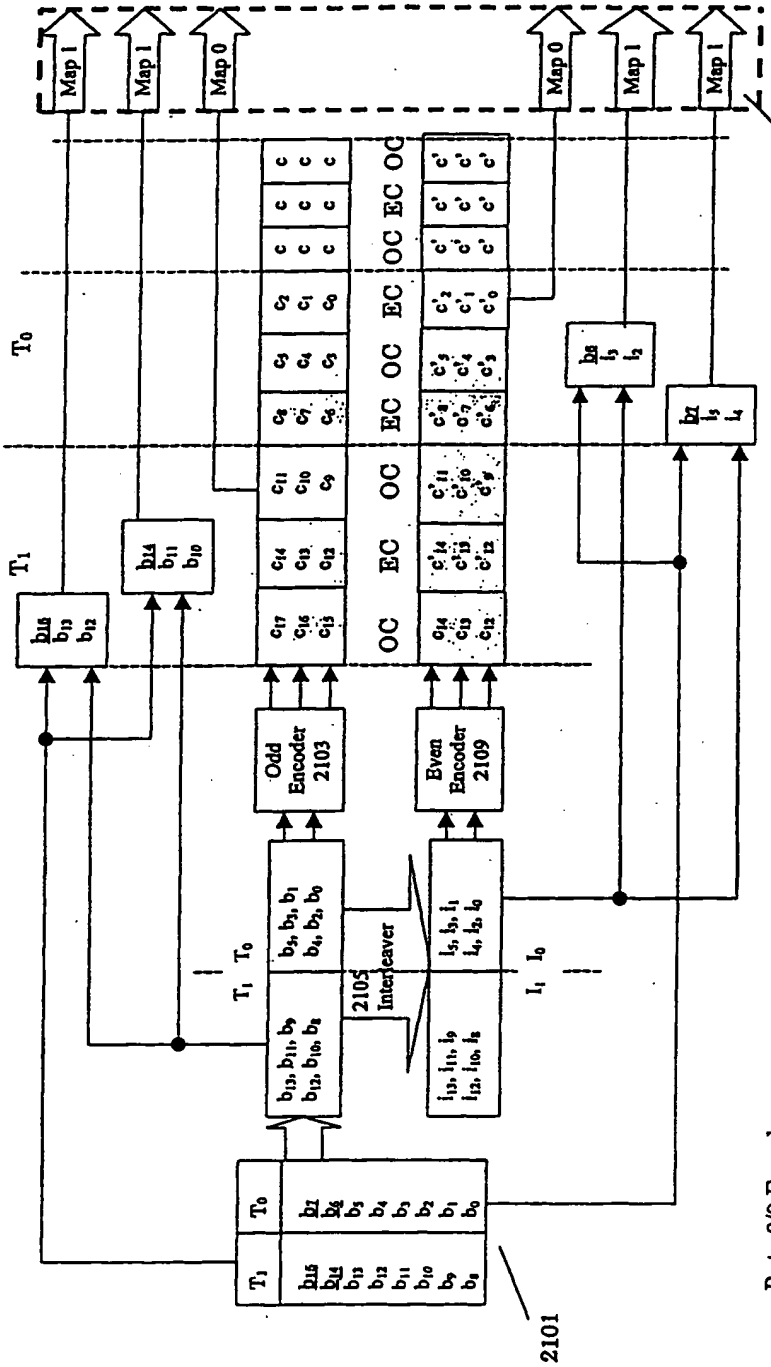

Map 1
Map 1
Map 0
Map 0
Map 1
Map 1
2111
OC EC OC
EC OC EC
OC EC OC
T0
T1
Odd Encoder 2103
Even Encoder 2109
2105 Interleaver
T1 T0
I1 I0
2101
Rate 8/9 Encoder

Figure 21A

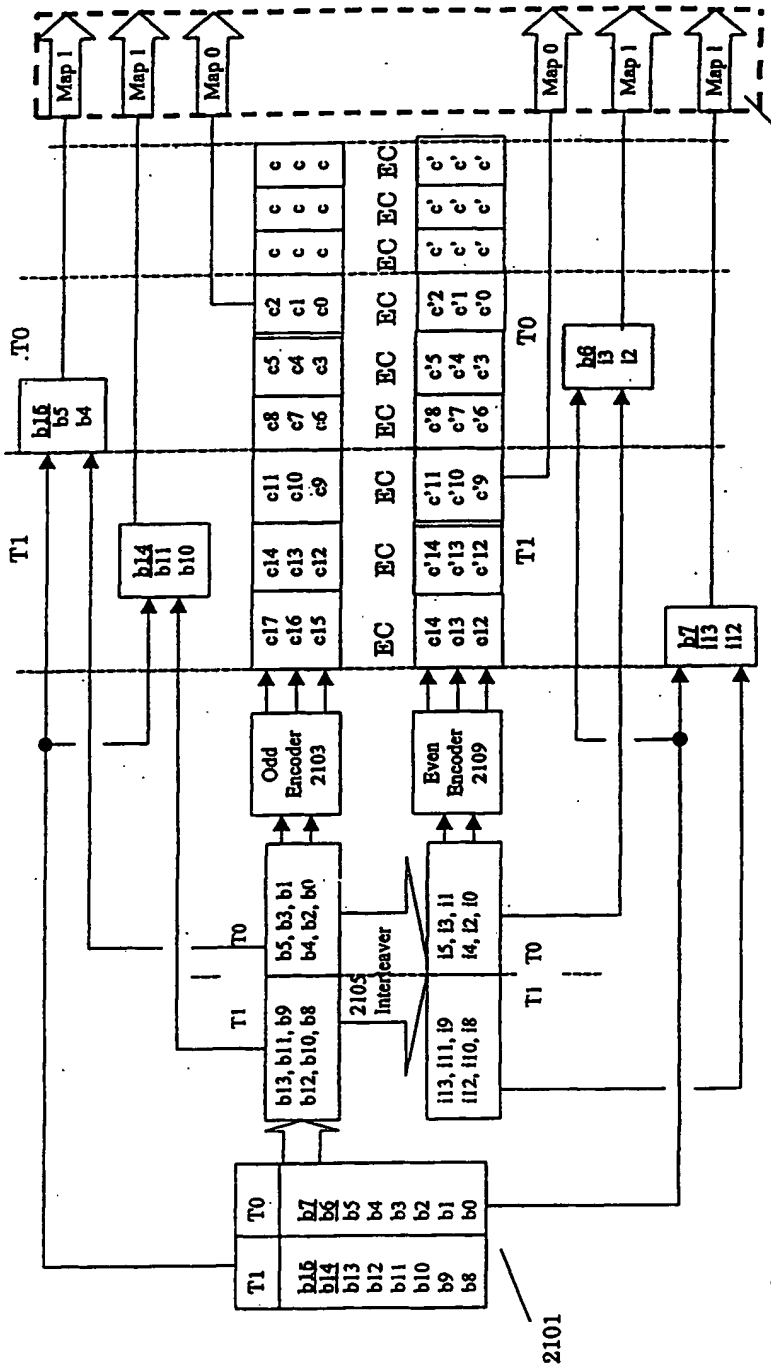

Map 1
Map 1
Map 0
Map 0
Map 1
Map 1
2111
T0
T1
EC
Odd Encoder 2103
Even Encoder 2109
2105 Interleaver
2101
Rate 8/9 Encoder (Alternate)

Figure 21B

Map 0

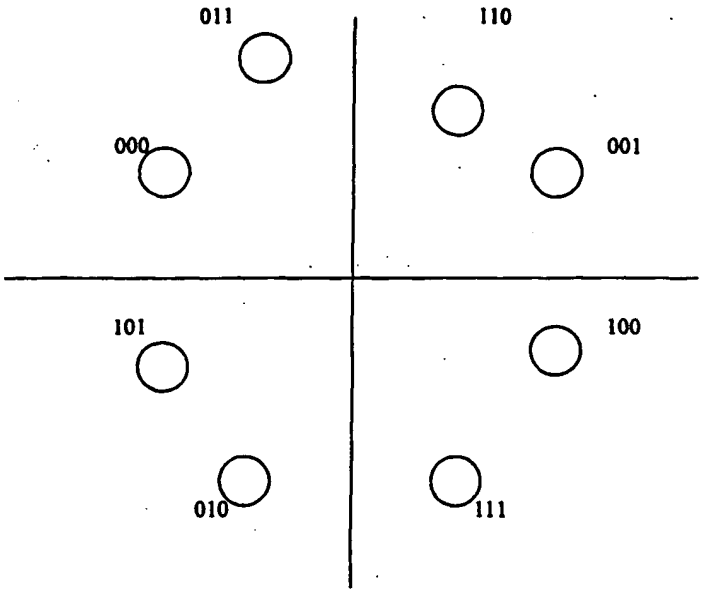


Figure 22

Map 1

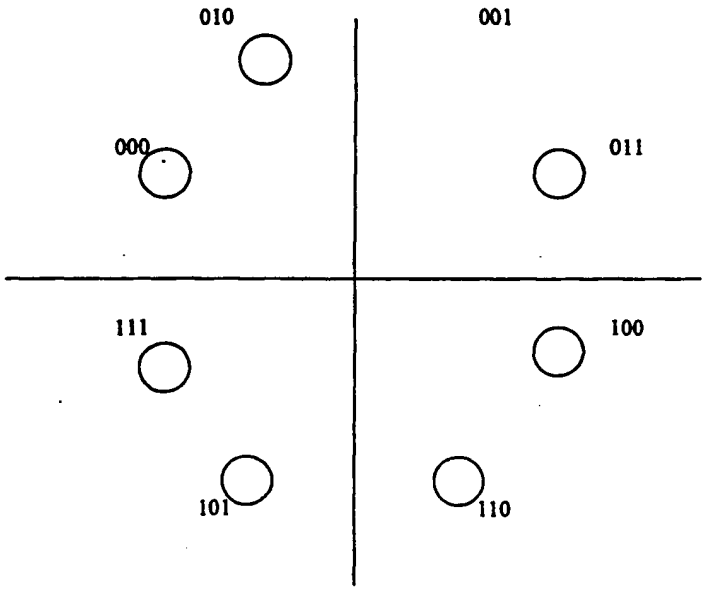


Figure 23

Map 2

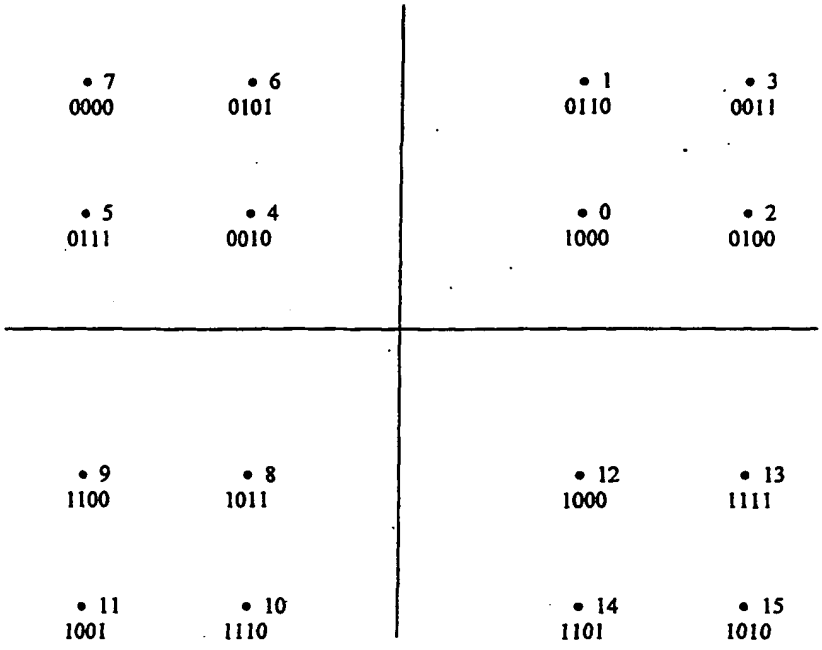


Figure 24

Map 3

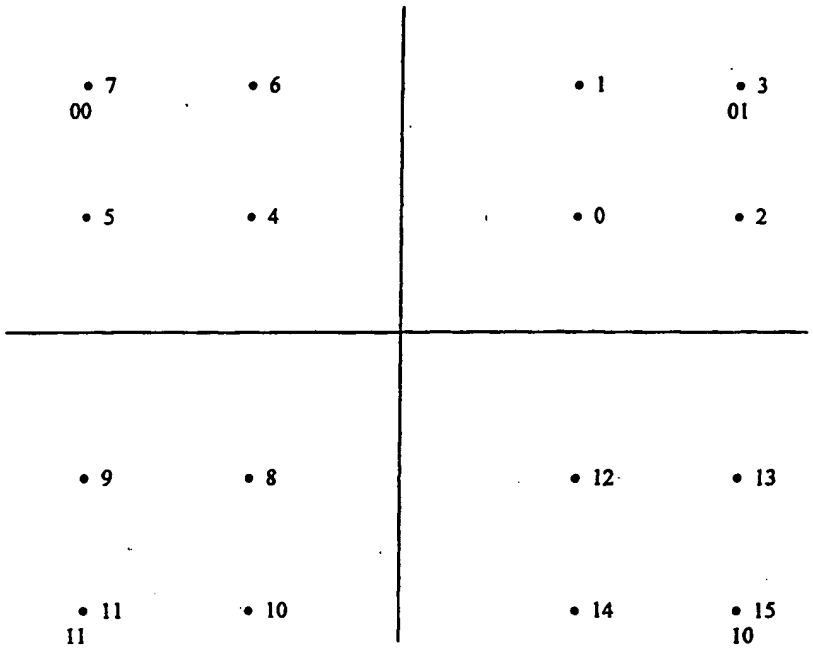


Figure 25

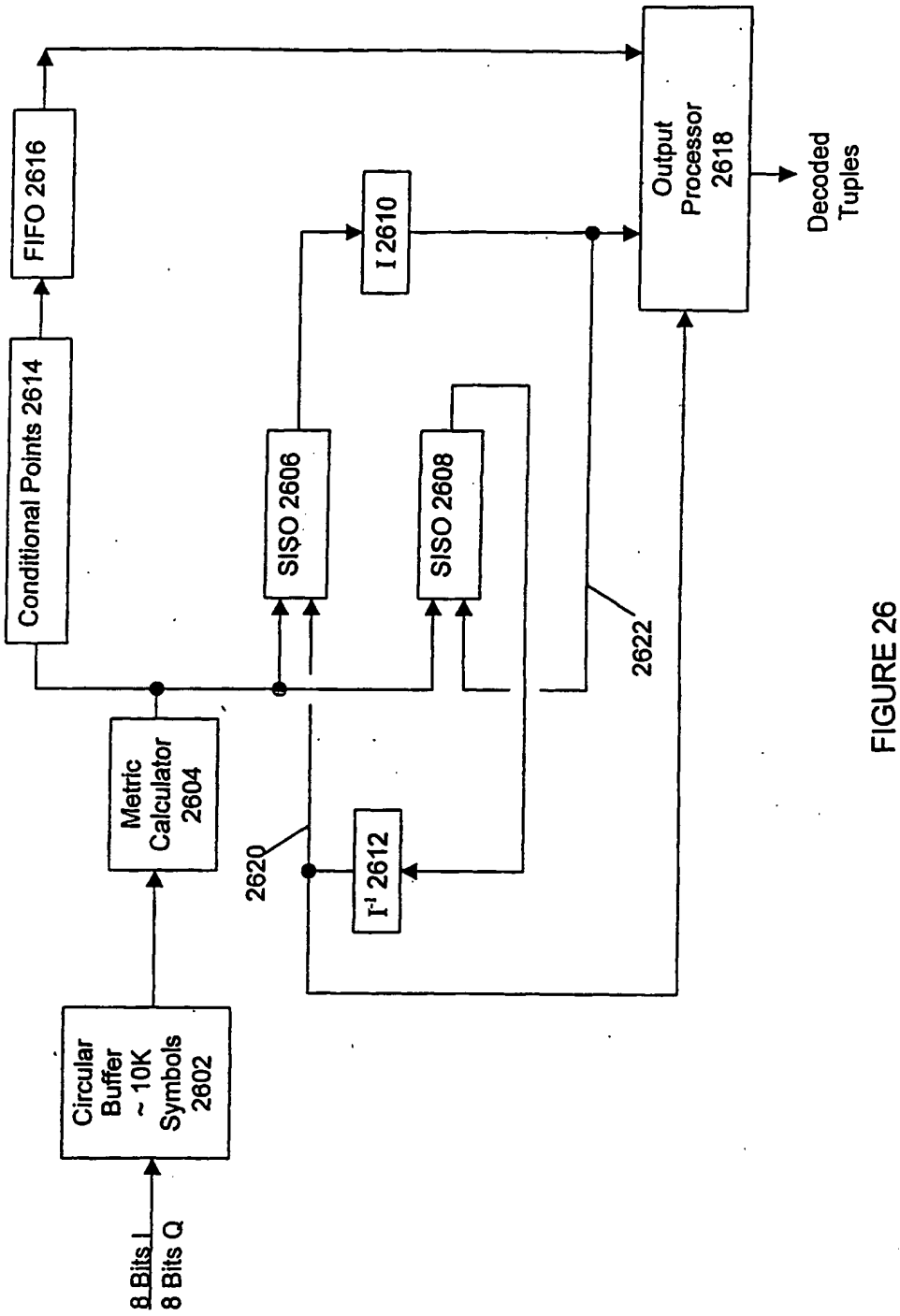

8 Bits I
8 Bits Q
Circular Buffer ~ 10K Symbols 2602
Metric Calculator 2604
Conditional Points 2614
FIFO 2616
SISO 2606
SISO 2608
I 2610
Γ^{-1} 2612
2620
2622
Output Processor 2618
Decoded Tuples

FIGURE 26

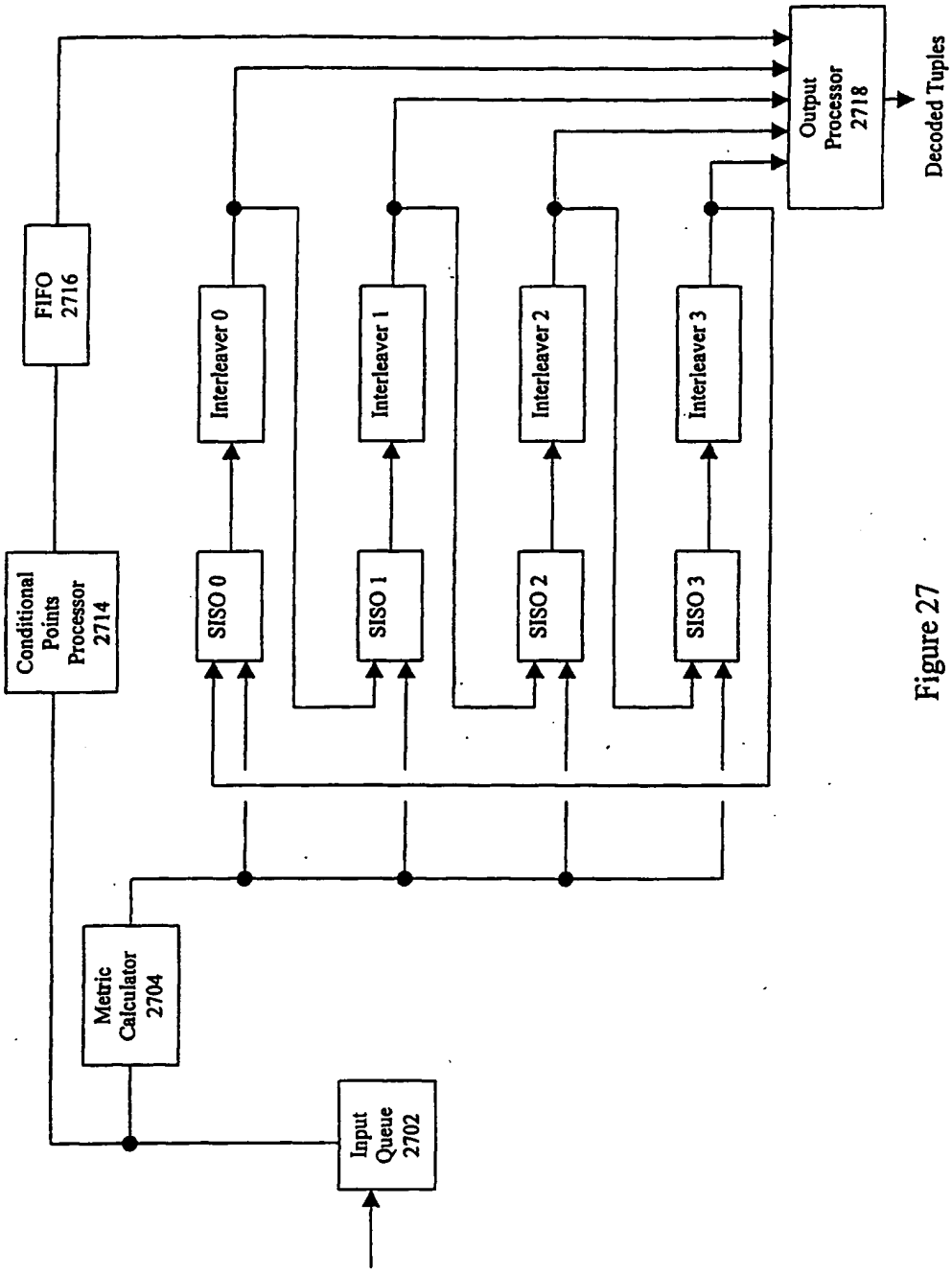

Output Processor 2718
Decoded Tuples
FIFO 2716
Interleaver 0
Interleaver 1
Interleaver 2
Interleaver 3
Conditional Points Processor 2714
SISO 0
SISO 1
SISO 2
SISO 3
Metric Calculator 2704
Input Queue 2702

Figure 27

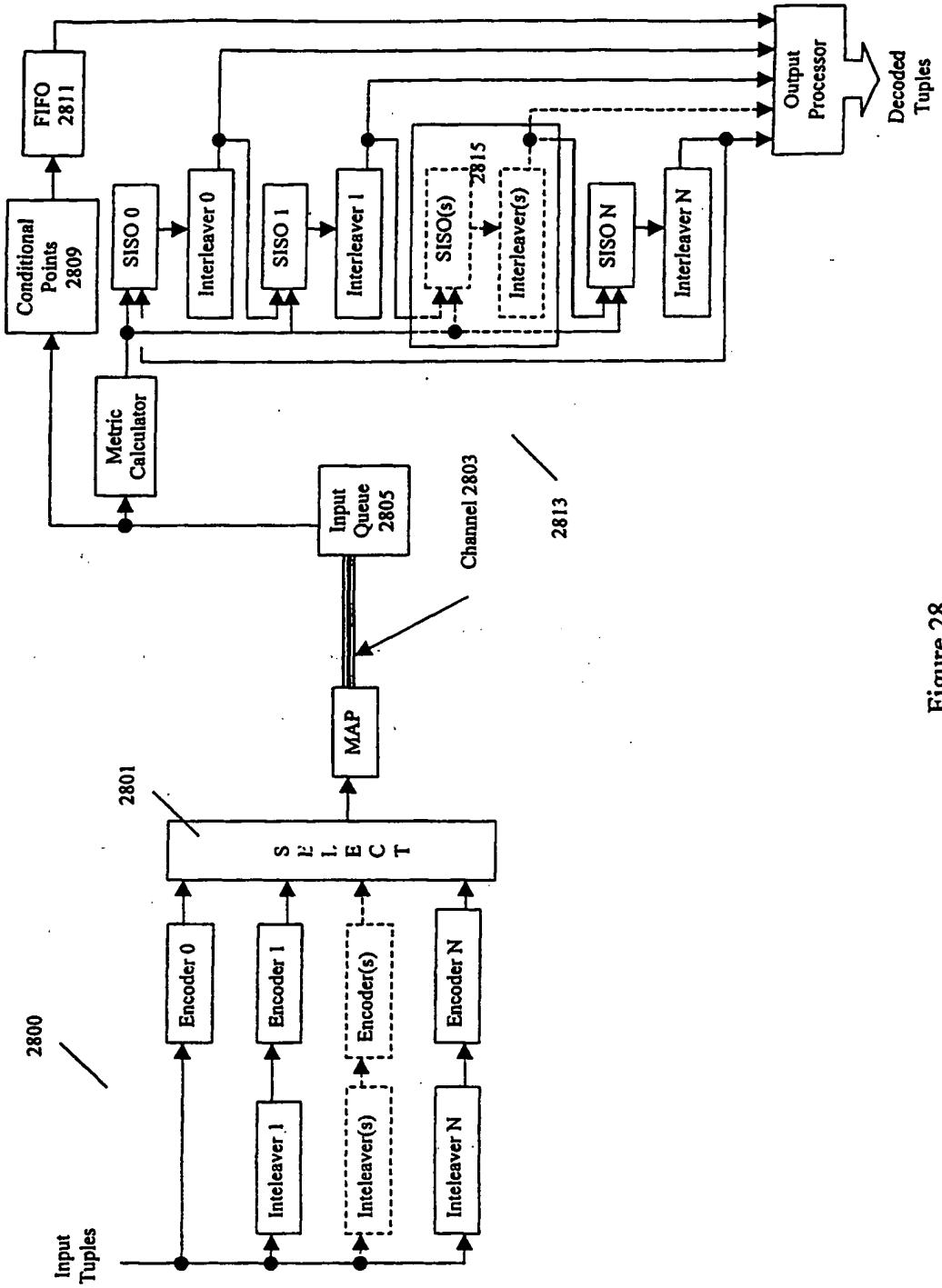

Input Tuples
2800
Encoder 0
Encoder 1
Encoder(s)
Encoder N
Inteleaver 1
Inteleaver(s)
Inteleaver N
2801
SELECT
MAP
Channel 2803
2813
Input Queue 2805
Metric Calculator
Conditional Points 2809
FIFO 2811
SISO 0
Interleaver 0
SISO 1
Interleaver 1
SISO(s)
Interleaver(s)
2815
SISO N
Interleaver N
Output Processor
Decoded Tuples

Figure 28

INTERNATIONAL SEARCH REPORT

Inter d Application No
PCT/US 01/27597

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H03M13/29 H03M13/27

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H03M H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CHUNG-WAI Y ET AL: "On the FER performance and decoding complexity of turbo codes" 1999 IEEE 49TH. VEHICULAR TECHNOLOGY CONFERENCE. HOUSTON, TX, MAY 16 - 20, 1999, IEEE VEHICULAR TECHNOLOGY CONFERENCE, NEW YORK, NY: IEEE, US, vol. 3 CONF. 49, 16 May 1999 (1999-05-16), pages 2214-2218, XP002150058 ISBN: 0-7803-5566-0	1,2,6,8, 9,14,15, 20,21
Y	the whole document	19

-/—

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 January 2002	28/01/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Ogor, M

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	FAZEL K ET AL: "Combined multilevel turbo-code with 8PSK modulation" GLOBAL TELECOMMUNICATIONS CONFERENCE, 1995. CONFERENCE RECORD. COMMUNICATION THEORY MINI-CONFERENCE, GLOBECOM '95., IEEE SINGAPORE 13-17 NOV. 1995, NEW YORK, NY, USA,IEEE, US, 13 November 1995 (1995-11-13), pages 649-653, XP010164368 ISBN: 0-7803-2509-5	1,2,6,8, 9,14,15, 20,21
A	the whole document	19
X	WO 99 19994 A (GHAZVINIAN FARZAD ;HINEDI SAMI M (US); STURZA MARK A (US); TELEDES) 22 April 1999 (1999-04-22) page 3, line 1 - line 7	1,2,6,8, 9,14,15, 20,21
A	page 4, line 10 - line 16	19
Y	EP 0 986 181 A (NDS LTD) 15 March 2000 (2000-03-15) the whole document	19
A	US 5 675 585 A (BONNOT CHRISTOPHE ET AL) 7 October 1997 (1997-10-07)	
A	HSU J-M ET AL: "A parallel decoding scheme for turbo codes" ISCAS '98. PROCEEDINGS OF THE 1998 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS. MONTEREY, CA, MAY 31 - JUNE 3, 1998, IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, NEW YORK, NY: IEEE, US, vol. 1, 31 May 1998 (1998-05-31), pages 445-448, XP002124484 ISBN: 0-7803-4456-1	
A	RAMSEY J L: "REALIZATION OF OPTIMUM INTERLEAVERS" IEEE TRANSACTIONS ON INFORMATION THEORY, IEEE INC. NEW YORK, US, vol. 16, no. 3, 1 May 1970 (1970-05-01), pages 338-345, XP002010849 ISSN: 0018-9448 cited in the application	

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte al Application No
PCT/US 01/27597

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9919994 A	22-04-1999	US 6157642 A	05-12-2000
		EP 1078489 A2	28-02-2001
		WO 9919994 A2	22-04-1999
EP 0986181 A	15-03-2000	EP 0986181 A2	15-03-2000
US 5675585 A	07-10-1997	FR 2723282 A1	02-02-1996
		CA 2154995 A1	30-01-1996
		EP 0695052 A1	31-01-1996
		NO 952969 A	30-01-1996